Neural Computing and Applications (2023) 35:2717–2735
https://doi.org/10.1007/s00521-022-07689-1

ORIGINAL ARTICLE

Knowledge-aware attentional neural network for review-based movie recommendation with explanations

Yun Liu[1] · Jun Miyazaki[1]

Received: 9 February 2022 / Accepted: 28 July 2022 / Published online: 4 September 2022

Abstract
In this paper, we propose a knowledge-aware attentional neural network (KANN) for dealing with movie recommendation tasks by extracting knowledge entities from movie reviews and capturing understandable interactions between users and movies at the knowledge level. In most recommendation systems, review information is already widely utilized to uncover the explicit preferences of users for items, especially for domains including movie recommendations, music recommendations, and book recommendations, as reviews are full of knowledge entities relevant to the domain. When processing review information, current methods usually use word embeddings to represent reviews for modeling users and items. As a result, they may split the meaning of a phrase, and thereby induce erroneous predictions. Moreover, most methods capture high-order interactions between users and items after obtaining latent low-dimensional representations, which means they cannot discover understandable interactions or provide knowledge-level explanations. By incorporating knowledge graph representation into movie recommendation tasks, the proposed KANN can not only capture the inner attention among user (movie) reviews but also compute the outer attention values between users and movies before generating corresponding latent vector representations. These characteristics enable the explicit preferences of users for movies to be learned and understood. We test our model on two datasets (IMDb and Amazon) for the movie rating prediction task and the click-through rate prediction task and show that it outperforms some of the existing state-of-the-art models and gains outstanding prediction performances in cases with a very small amount of reviews. Furthermore, we demonstrate the high explainability of the proposed KANN by visualizing the interaction between users and movies through a case study. Our results and analyses highlight the relatively high effectiveness and reliability of KANN for movie recommendation tasks.

Keywords Attention neural networks · Explainable recommendation · Review-based recommender systems · Knowledge graph · Personalization

1 Introduction

Recommender systems (RSs) are critical for many online services in terms of liberating users from the ever-growing number of choices. Thus, more and more studies have been studying RSs with explanations [1, 2]. A transparent RS with explanations has the potential to increase the trust of users and to help them choose appropriate items faster [2]. Obviously, reviews reflect user preferences and reveal rich properties of items that cannot be conveyed via ratings. These preferences of users and properties of items can thus be used to present explainable recommendations to users. In addition, the rich features of reviews can be used to solve the data sparsity and cold start issues caused by rare ratings [3]. Therefore, there has been immense interest in review-based recommendations to improve recommendation performances and to provide explanations [1, 4, 5].

Recent works have proposed using word embeddings of reviews as input for training neural network models and providing review-level [4–6] or word-level [7] explanations via an attention mechanism. These works follow a similar pattern to generate explainable recommendations

✉ Jun Miyazaki
miyazaki@c.titech.ac.jp

Yun Liu
liu@lsc.c.titech.ac.jp

[1] Department of Computer Science, School of Computing, Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo 152-8550, Japan

from reviews. First, they concatenate the word embeddings of user (item) reviews to represent a user (item). Then, a neural network processes the review embeddings and learns a single latent vector for the user (item). At the same time, informative words or reviews are selected in accordance with the high-weight values generated by the attention mechanism. Finally, feature interactions within a user-item pair are obtained by a fixed-dimension vector for the user (item), which is generated by compressing reviews.

The above description reveals two key aspects of a review-based explainable RS: first, it extracts meaningful and effective features from reviews, and second, it captures the interactions between users and items for learning important features from reviews that can model the preferences of users for items. These two points also bring about two challenges in recommendation using online reviews.

First, many works have adopted convolutional neural networks (CNNs) to process reviews [5–7]. Although CNNs can keep short sequence information and discern informative words, their ability to model noisy, fragmented, and long-tailed review data is limited. Moreover, when using word embeddings, the semantic expression quality of low-frequency words decreases more significantly than that of high-frequency words [3]. In addition, the word-level explanations may split the meaning of the phrase and cause misunderstanding, which could lead to users having difficulty understanding the meaning of highlighted words unless they have also read the contextual information of the review or the entire review itself.

As for the second challenge, capturing different-order feature interactions between users and items is burdensome. Most current works have focused on extracting high-order feature interactions from word embeddings by using a neural network and then capturing user-item interactions after obtaining latent semantic vectors [6, 8]. Although these methods can generate implicit feature combinations effectively, they barely explore the abundant interaction information at a lower level [5, 6]. Tay et al. [4] considered low-order relations based on word embedding features, but since they chose to sum all word embeddings of a review into a low-dimensional vector as an input vector, this processing method unavoidably degrades the modeling ability of a user (item) and leads to erroneous predictions.

Recently, knowledge graphs (KGs) have been successfully applied in scenarios of content-based recommendation [9], machine reading [10], and text classification [11]. In general, reviews in specific domains such as movie recommendations, music recommendations, and book recommendations comprise many knowledge entities and a lot of common sense. Considering the challenges of a review-based RS mentioned above and inspired by the success of using KGs, we propose a knowledge-aware neural network (KANN) to predict movie review ratings and provide knowledge-level explanations. To avoid semantic error problems caused by splitting phrases and low-frequency words, we first extract knowledge features from reviews by introducing a KG. Then, we learn knowledge embeddings from the KG in view of the transitivity of correlation and the exploration of new features. Subsequently, we explore the inner interaction between different knowledge embeddings of a user (movie) and capture the outer interaction between the pair of a user and a movie by means of an attention mechanism. As a result, the interaction learning can capture the relation between any two features, regardless of the distance between them. Afterward, we also capture high-order interactions between a user and a movie in accordance with low-dimension representations of latent knowledge. Finally, we visualize the interaction weights and provide knowledge-level explanations through a case study.

The main contributions of this paper are as follows.

- Our proposed KANN can extract knowledge features from reviews to represent users and movies while not only avoiding semantic noise but also decreasing the number of input features in comparison with popular models using the word embedding method.
- To the best of our knowledge, we are the first to model interactions between a user and a movie at the knowledge-level before getting their latent vectors from reviews, while selecting informative entities.
- The experimental results demonstrate that KANN achieves a better performance for rating prediction tasks and CTR tasks than state-of-the-art methods. A case study of visualization also validates its effectiveness and explainability.

In Sect. 2 of this paper, we introduce the works most closely related to our proposed model. In Sect. 3, the four steps of knowledge feature extraction are described. In Sect. 4, the framework of KANN is introduced. In Sect. 5, we present the prediction performances of the proposed model in comparison with other existing models. In Sect. 6, we discuss our analyses of the proposed model. We conclude in Sect. 7 with a brief summary and mention of future work.

2 Related works

In this section, we introduce existing works in three areas that are highly relevant to our own.

2.1 Review-based recommendation

Many studies have exploited reviews to improve recommendation performances [1, 2, 4, 12, 13]. These approaches have not only alleviated the cold-start problem caused by sparse relations between users and items but also explored the semantic representation of users and items. Some of these works focused on the topic modeling or sentiment classification of reviews to better understand user behaviors and item properties [1, 2, 14, 15]. For example, Diao et al. [14] proposed representing user preferences and movie properties by applying aspect-specific sentiment words uncovered by topic modeling. Other works have exploited the use of neural networks for learning the latent semantic representations of users and items from reviews [4–6]. Catherine et al. [6] and Zheng et al. [16] introduced a CNN as an extractor to encode user reviews and item reviews in parallel, before matching the user and the item by feeding their outputs into a factorization machine [17]. Almahairi et al. [18] proposed a language regularized latent factor (LMLF) model that adopts an RNN to regularize the collaborative filtering matrix factorization.

The works on review-based recommendation not only consider item properties but also extract user preferences in reviews, which helps to improve the recommendation performance. These works, however, have three limitations. First, topic-based and sentiment-based models usually require expert knowledge to define or identify topics or sentiments in reviews [2, 19]. Second, CNN-based models usually ignore the long-distance dependencies between different words, which may result in the meaning being cut out of phrases and the introduction of noise. Third, most review-based models only focus on high-order interactions between users and items, while ignoring the extraction of understandable interactions based on low-order user-item relations.

2.2 Attention model with explanations

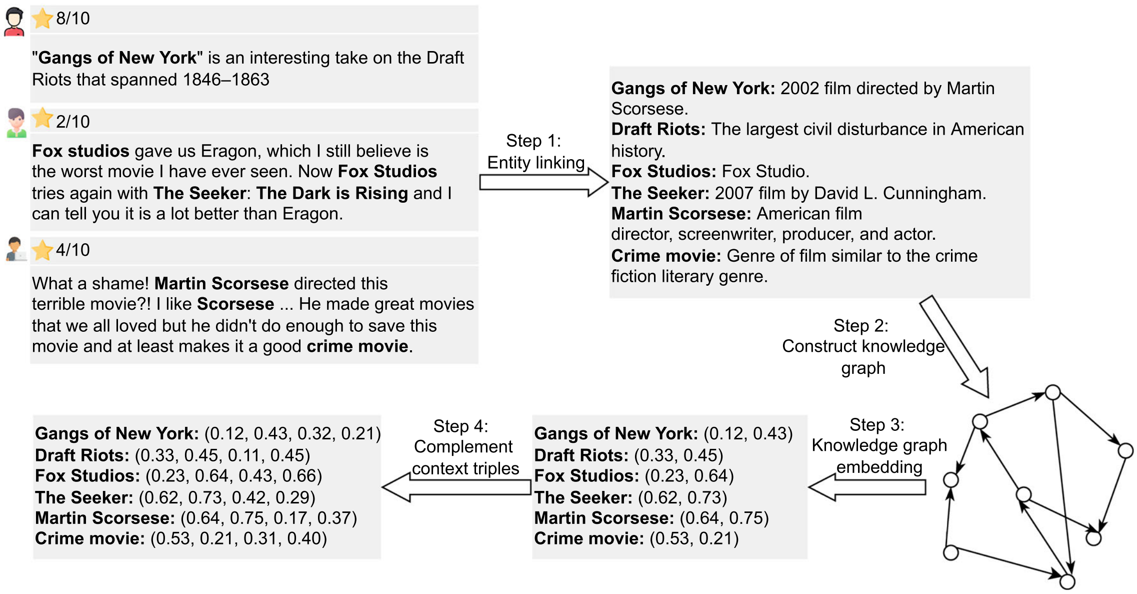
The attention mechanism has recently been widely applied in natural language processing [20, 21]. For machine translation, Bahdanau et al. [20] proposed an attention-based encoder-decoder architecture to select the parts of a source sentence that are relevant to a target word. In another work, a transformer relies solely on the self-attention to reduce sequential computation [21] and encodes position information as one part of the input in machine translation to maintain the word order. In the RS field, Seo et al. [7] proposed a model that learns user preferences and item properties from reviews through a CNN with dual local and global attention. Chen et al. [5] presented neural attentional rating regression with review-level explanations (NARRE) to explore the usefulness of reviews. Tay et al. [4] proposed a multi-pointer co-attention network (MPCN) method that adopts hard attention to extract the most informative review of each user and item, and then uses co-attention to model the word-level interaction between the matched reviews.

The review-based attention models in the RSs field can identify the important reviews or words as the explanations of the corresponding recommendation results. While these models have greatly improved the transparency of their systems, and are able to provide convincing recommendation results to users, they also have their limitations. Tay et al. [4] proposed summing all the word embeddings of a review into a short vector to represent the review and then concatenating all reviews to model a user or an item. However, this method only selects a fixed number out of all reviews, which may result in important features being ignored. Chen et al. [5] proposed a method that jointly models rating and review information to select informative reviews, but it cannot identify the contribution of reviews for recommendation results.

2.3 Knowledge graph for recommendation

KGs have been widely utilized in recommendation [9, 22–25]. Some works have used them to extract latent knowledge-level connections for content-based recommendation. Wang et al. [9] extracted knowledge embeddings from news titles and fused semantic representations for click-through rate prediction. Cheekula et al. [26] introduced a KG named DBpedia to search movie-relevant entities in accordance with a movie identifier in Movielens. They then used a spreading activation algorithm to identify personalized entities for recommendation. Zhu et al. [27] built a dedicated KG based on manually defined relations and then incorporated the users' clicked history sequences and pre-searched paths between users and items in the KG for improving recommendation accuracy. Other works only extract the knowledge entity corresponding to the item. Wang et al. [28] proposed a multitask feature learning approach for KG-enhanced recommendation, where two tasks are used to jointly learn ratings and entities for click-through rate prediction. Wang et al. [23] proposed a path-based knowledge-aware model to search for a path starting from the item corresponding to an entity for the click-through rate prediction (CTR) task. The methods with knowledge explanations usually provide interpretation results by defining meta-paths [29, 30] or learning propagation paths[31] in a KG. Ma et al. [30] proposed a rule-guided neural recommendation method that mines inductive rules from an item-centric KG. Wang et al. [31] distilled paths carrying the high-order relation information from the item-side KG.

The technology of integrating the KG into the recommendation provides a new perspective for improving the performance and explainability of RSs. However, current knowledge-based works mainly focus on the connections



"**Gangs of New York**" is an interesting take on the Draft Riots that spanned 1846–1863

Fox studios gave us Eragon, which I still believe is the worst movie I have ever seen. Now **Fox Studios** tries again with **The Seeker**: **The Dark is Rising** and I can tell you it is a lot better than Eragon.

What a shame! **Martin Scorsese** directed this terrible movie?! I like **Scorsese** ... He made great movies that we all loved but he didn't do enough to save this movie and at least makes it a good **crime movie**.

Gangs of New York: 2002 film directed by Martin Scorsese.
Draft Riots: The largest civil disturbance in American history.
Fox Studios: Fox Studio.
The Seeker: 2007 film by David L. Cunningham.
Martin Scorsese: American film director, screenwriter, producer, and actor.
Crime movie: Genre of film similar to the crime fiction literary genre.

Gangs of New York: (0.12, 0.43)
Draft Riots: (0.33, 0.45)
Fox Studios: (0.23, 0.64)
The Seeker: (0.62, 0.73)
Martin Scorsese: (0.64, 0.75)
Crime movie: (0.53, 0.21)

Gangs of New York: (0.12, 0.43, 0.32, 0.21)
Draft Riots: (0.33, 0.45, 0.11, 0.45)
Fox Studios: (0.23, 0.64, 0.43, 0.66)
The Seeker: (0.62, 0.73, 0.42, 0.29)
Martin Scorsese: (0.64, 0.75, 0.17, 0.37)
Crime movie: (0.53, 0.21, 0.31, 0.40)

Fig. 1 Knowledge feature extraction process based on three reviews of the movie "Gangs of New York" on IMDb. For the first part, each user provides one rating and one review, where the bold words in each review are mentions corresponding to the knowledge entities. The second part contains the knowledge entities extracted from reviews and their descriptions. The third part is a KG constructed on the basis of knowledge entities. The fourth and fifth parts are examples of the representations of each knowledge entity

between knowledge entities with items, or the connections based on their own defined item-item interactions, which induce limited knowledge information in their KGs. Moreover, they generally extract knowledge entities from the content description of an item or identifier information but ignore the explicit connections between users and knowledge entities contained in reviews.

3 Knowledge feature extraction

In this section, we describe the process of extracting knowledge features from reviews based on a KG. The extracted knowledge features are used as the input of our proposed model. Figure 1 illustrates the detailed extraction process with an example from the IMDb dataset (described later in Sect. 5). The overall process consists of four steps. First, we apply the entity linking method [32, 33] to identify mentions of reviews by connecting them with entities in a KG named Wikidata.[1] Second, after obtaining these review entities, we construct a directed KG by searching for the relationships between entities in Wikidata. To avoid the cold-start problem caused by sparse data, we construct a subgraph by looking for all the relation links of each review entity within a one-hop distance (having a direct connection with the review entity) in Wikidata. Third, given the extracted KG, we use TransE [34] to learn the representations of entities and links. The learned embedding of a single entity is fed into the subsequent recommendation model [9]. Lastly, we explore the contextual entities of the review entities in the KG, where the contextual entities are the immediate neighbors of each review entity. Unlike the operation in [9], we not only explore context entities but also consider the type of relation between an entity and its context entities.

In a directed subgraph, a directed link represents a relation, and a node represents an entity. An entity directed by another entity via a relation is called an object, while another entity is called a subject. The combination of a subject, a relation, and an object is called a triple. The context information of entity e is defined as a set of triples. In these triples, the entity is either a subject or an object:

$$\mathrm{context}(e) = \{t_i \mid t_i = (e, r, e_i) \parallel t_i = (e_i, r, e), t_i \in G\}, \tag{1}$$

where G is the constructed knowledge subgraph, t_i and e_i are a context triple and an entity of e, respectively, and r is the relation between e and e_i. The representation vectors of e, e_i, and r are embeddings learned by TransE [34]. We used the pretrained embeddings of Wikidata with TransE in the OpenKG framework [35]. To be more specific, the *TransE* method learns each entity and relation by optimizing the translation principle $\boldsymbol{e} + \boldsymbol{r} \approx \boldsymbol{e}_i$ when entity e is a head entity and optimizing $\boldsymbol{e}_i + \boldsymbol{r} \approx \boldsymbol{e}$ when e is a tail entity, where $\boldsymbol{e}$, $\boldsymbol{e}_i$, and $\boldsymbol{r}$ are the embeddings of entity e, e_i, and relation r, respectively.

[1] https://www.wikidata.org/wiki/Wikidata:Main_Page.

To preserve the information of an entity in a graph as accurately as possible, we add the embedding of a subject, a relation, and an object as the embedding of a triple. The context embedding of entity e, $\bar{\mathbf{e}}$, is calculated as the average of all its context triples:

$$\bar{\mathbf{e}} = \frac{1}{|\text{context}(e)|} \sum_{t_i \in \text{context}(e)} \boldsymbol{t_i}, \tag{2}$$

where $\boldsymbol{t_i}$ is the embedding of context triple t_i. Since we use both the context triples and the entity itself to express the entity, it makes sense to combine the two kinds of embeddings by concatenating them. The final representation vector of an entity is thus denoted as

$$\mathbf{e} = \text{concat}([\bar{\mathbf{e}}, \hat{\mathbf{e}}], axis = 1), \tag{3}$$

where $\hat{\mathbf{e}}$ is the entity embedding learned by TransE [34] from a KG (corresponding to step 3 in Fig. 1), concat() is a concatenation operation, and axis $= 1$ means we concatenate the two vectors along the column.

The representations of all the review entities in the KG can be formulated as matrix E, and

$$E = \text{concat}([\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_i, ...], axis = 0), \tag{4}$$

where $E \in \mathbb{R}^{n \times d}$, n is the number of entities in the KG, d is the dimension of vector $\mathbf{e}_i$, $\mathbf{e}_i$ is calculated by Eq. (3), and $axis = 0$ means that these vectors are concatenated along the row.

4 Knowledge-aware attentional neural network

In this section, we introduce our proposed knowledge-aware attentional neural network (KANN) model. First, we present its general architecture and the problem formulation of our work in Sect. 4.1. Then, we elaborate on our low-order attention-based interaction layer in Sect. 4.2. Lastly, we describe the prediction layer in Sect. 4.3 and our training procedure in Sect. 4.4.

4.1 Overview of KANN

The objectives of our model are to predict the ratings of movies corresponding to each user and to give a reasonable explanation of each user-movie pair by modeling the low-order knowledge feature interaction between the user and the movie. The left part in Fig. 2 shows a user-movie pair as an example to illustrate the architecture of KANN, and the middle and right parts are illustrations of the inner and outer attention in our model, respectively. We use two parallel neural networks to deal with the reviews of the user and the movie separately. After obtaining the knowledge embeddings of a user and a movie through knowledge feature extraction, we use an inner-attention mechanism to capture the context information in the reviews of the user (movie). Then, we use an outer-attention mechanism to model the low-order interaction between the user and the movie from the reviews. For the prediction layer, we capture the high-order interaction between the user and the movie by using their latent factors, and then calculate the final rating.

We formulate the rating prediction problem based on reviews as follows. For a given user i, the review entities are represented as $\mathcal{U}_i = \{e_i^1, e_i^2, ..., e_i^k, ...e_i^l\}$, and the corresponding embedding matrix is calculated based on a look-up operation on E, which is denoted as $U_i = (\mathbf{e}_i^1, \mathbf{e}_i^2, ..., \mathbf{e}_i^k, ...\mathbf{e}_i^l)$, where $U_i \in \mathbb{R}^{l \times d}$ represents an embedding matrix of reviews written by the user i. Here, l is the length of the entities extracted from the reviews of user i, and d is the dimension of the embedding for the entity $\mathbf{e}_i^k$. Similarly, we denote the review entities and the corresponding embedding matrix of reviews for a movie j as $\mathcal{M}_j$ and $M_j \in \mathbb{R}^{l \times d}$, respectively. Note that the two parallel neural networks in Fig. 2 perform the same modeling operations on reviews of users and movies. In the following descriptions, we introduce the process of modeling reviews corresponding to users.

4.2 Attention-based review interaction

The purpose of using the attention mechanism in our model is to select knowledge entities that can represent both the user preferences and the movie properties. It contains two layers: an inner-attention layer and an outer-attention layer. The inner-attention layer captures the inner relationships of user (movie) review entities in the knowledge semantic space, which can control the importance of some entities in accordance with their connections with others. The outer-attention layer calculates the attention score between the user and the movie at the level of knowledge entities. The representation vector of user reviews obtained from these two layers includes the context knowledge information of the user reviews and the corresponding knowledge information of the movie reviews closely related to this user.

We project the knowledge embedding features of the user i into H knowledge semantic spaces by a linear projection, which can be formulated as

$$U_{ih} = U_i W_p, \tag{5}$$

where $U_{ih} \in \mathbb{R}^{l \times d_{\text{model}}}$ is the review representation matrix of user i in space h, $W_p \in \mathbb{R}^{d \times d_{\text{model}}}$ is a projection matrix of space h, d_{model} is the dimension of the latent embedding after the linear projection, and $d = H * d_{\text{model}}$. The

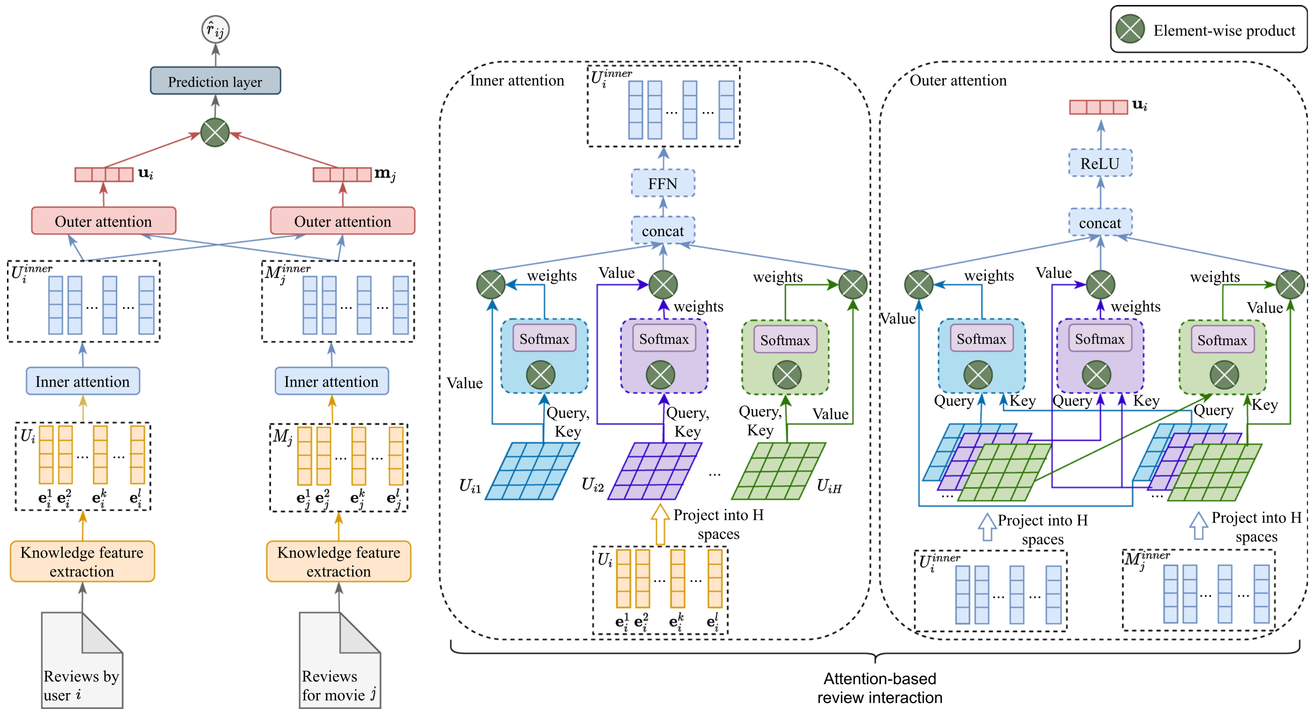


Fig. 2 Architecture of proposed knowledge-aware attentional neural network (KANN)

effectiveness of the linear projection in Eq. (5) has been proven to improve prediction performance [36].

For the inner attention, given the review representation matrix of user i, $\{U_{i1}, U_{i2}, ..., U_{ih}, ..., U_{iH}\}$, we apply an inner-attention function on H spaces at the same time. This is because an attention mechanism can be described as mapping a query and a set of key-value pairs to an output [21]. Therefore, for space h, we project U_{ih} into queries Q_{ih}, keys K_{ih}, and values V_{ih} with three linear projection matrices parameterized by W_{qh}, W_{kh}, and W_{vh}, as

$$Q_{ih} = U_{ih} W_{qh}, \tag{6}$$

$$K_{ih} = U_{ih} W_{kh}, \tag{7}$$

$$V_{ih} = U_{ih} W_{vh}, \tag{8}$$

where $W_{qh}, W_{kh}, W_{vh} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, and $Q_{ih}, K_{ih}, V_{ih} \in \mathbb{R}^{l \times d_{\text{model}}}$.

We calculate the inner-attention output matrix U_{ih}^{inner} as

$$U_{ih}^{\text{inner}} = \text{softmax}\left(\frac{Q_{ih} K_{ih}^T}{\sqrt{d_{\text{model}}}}\right) V_{ih}, \tag{9}$$

where $U_{ih}^{\text{inner}} \in \mathbb{R}^{l \times d_{\text{model}}}$. These H outputs from different spaces are concatenated and reorganized to obtain

$$U_i^{\text{inner}} = \text{FFN}(\text{concate}(U_{i1}^{\text{inner}}, U_{i2}^{\text{inner}}, ..., U_{ih}^{\text{inner}}, \ldots, U_{iH}^{\text{inner}})), \tag{10}$$

where $\text{FFN}(\cdot)$ is a feed forward network with ReLU activation, and $U_i^{\text{inner}} \in \mathbb{R}^{l \times d}$ is the representation matrix of user i after the inner-attention operation.

Likewise, we can obtain $M_j^{\text{inner}} \in \mathbb{R}^{l \times d}$ as the representation matrix of movie j after the inner-attention operation by

$$M_j^{\text{inner}} = F_{\text{inner}}(M_j), \tag{11}$$

where F_{inner} indicates the inner-attention operation, which is the same as the operations described in Eqs. (5)–(10).

We can also project the features of outer-attention into multiple spaces, the same as in the inner-attention case, so an outer-attention score matrix is also easy to derive. Therefore, we directly calculate the query matrix $\hat{Q}_i$ corresponding to the representation matrix U_i^{inner} of user i, and the key matrix $\hat{K}_j$ corresponding to the representation matrix M_j^{inner} of movie j, as

$$\hat{Q}_i = U_i^{\text{inner}} \hat{W}_q, \tag{12}$$

$$\hat{K}_j = M_j^{\text{inner}} \hat{W}_k, \tag{13}$$

where $\hat{W}_q \in \mathbb{R}^{d \times d}$ and $\hat{W}_k \in \mathbb{R}^{d \times d}$ are linear projection matrices. Based on Eqs. (12) and (13), the outer-attention matrix A can be represented as

$$A_{ij} = \mathrm{softmax}\left(\frac{\hat{Q}_i \hat{K}_j^T}{\sqrt{d_{\mathrm{model}}}}\right). \tag{14}$$

Then, we obtain the final representation vector of user i by using a nonlinear activation function to deal with the output matrix of the attention layer:

$$\mathbf{u}_i = ReLU\left(\mathbf{w}_u A_{ij} M_j^{\mathrm{inner}} \hat{W}_v + \mathbf{b}_{\mathbf{u}}\right), \tag{15}$$

where $\mathbf{u}_i \in \mathbb{R}^d$, $\mathbf{w}_u \in \mathbb{R}^l$, $\mathbf{b}_{\mathbf{u}} \in \mathbb{R}^d$, and $\hat{W}_v \in \mathbb{R}^{d \times d}$ is the projection matrix used for projecting the movie representation matrix into the same semantic space as for the user representation vector. Likewise, we can obtain $\mathbf{m}_j$ as the final representation vector of movie j.

4.3 Prediction layer

We calculate the interaction vector $\mathbf{h} \in \mathbb{R}^d$ by combining user i and movie j as follows:

$$\mathbf{h} = \mathbf{u}_i \odot \mathbf{m}_j, \tag{16}$$

where the operator $\odot$ denotes the element-wise product, and $\mathbf{u}_i$ and $\mathbf{m}_j$ are the output vectors of the outer attention layer corresponding to user i and movie j. We consider the bias information in the final prediction, and the final review rating of user i corresponding to movie j is obtained as follows:

$$\hat{r}_{ij} = \mathbf{w}_h \mathbf{h} + b_u + b_m + \mu, \tag{17}$$

where $\mathbf{w}_h \in \mathbb{R}^d$ is the weight matrix in the prediction layer, and $b_u \in \mathbb{R}$, $b_m \in \mathbb{R}$, and $\mu \in \mathbb{R}$ represent the user bias, movie bias, and global bias, respectively.

4.4 Training procedure of KANN

We leveraged two different tasks, a rating prediction task and a click-through rate (CTR) task, to evaluate the effectiveness of our proposed model.

4.4.1 Rating prediction task

The rating prediction task is a regression problem. We use the squared loss [5] as the loss function and denote it by

$$J_{rp} = \sum_{i,j} \left(\widehat{r}_{ij} - r_{ij}\right)^2, \tag{18}$$

where $\widehat{r}_{ij}$ is the predicted rating from user i to movie j, and r_{ij} is the ground-truth rating from user i to movie j.

4.4.2 CTR task

The CTR task is a classification problem. In this task, we can view the value of r_{ij} as a label, where 1 means user i likes movie j, and 0 otherwise. The loss function is the widely utilized sigmoid cross-entropy loss [36], denoted by

$$J_{ctr} = -\sum_{i,j} r_{ij} \log \sigma\left(\widehat{r}_{ij}\right) + \left(1 - r_{ij}\right) \log\left(1 - \sigma\left(\widehat{r}_{ij}\right)\right). \tag{19}$$

```
Algorithm 1 Training procedure of KANN
Input: entity embedding matrix $E$ of the KG,
  review entities of S users and N movies
  $[\mathcal{U}_1, \mathcal{U}_2, ..., \mathcal{U}_i, ..., \mathcal{U}_S]$, $[\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_j, ..., \mathcal{M}_N]$.
  for $epoch \leftarrow 1$ to $epochs$ do
      for $i \leftarrow 1$ to $S$ do
          $U_i = embedding\_lookup(E, \mathcal{U}_i)$
          inner attention by Eqs. (5)–(10)
      end for
      for $j \leftarrow 1$ to $N$ do
          $M_j = embedding\_lookup(E, \mathcal{M}_j)$
          inner attention by Eq. (11)
      end for
      outer attention by Eqs. (12)–(15)
      predict the ratings by Eqs. (16) and (17)
      minimize the loss by Eq. (18) or (19)
  end for
```

In our proposed method, the review features of a user and a movie are obtained by a knowledge feature extraction operation. Then, two attention layers are used to model the inner interactions in the user (movie), and the outer interactions between the user and the movie. Finally, a nonlinear activation function and the bias information are used to predict the final rating. The corresponding pseudo-code of our method is provided in Algorithm 1.

5 Experiments

In this section, we first give a brief description of the datasets. We then introduce the comparative baseline models, and list the corresponding parameter settings. Lastly, we report the prediction performances on standard datasets and on datasets with different sparsities.

5.1 Datasets

We conducted experiments with two publicly accessible datasets: the IMDb dataset and Amazon's "movies and TV" dataset.

1. **IMDb**. This movie rating dataset was published as part of Jointly Modeling Aspects, Ratings and Sentiments (JMARS) [14], which uncovered aspects and

Table 1 Basic statistics of review datasets and extracted KGs

Statistics	IMDb	Amazon
# users	2088	244,782
# movies	4668	59,652
# reviews	126,874	1,588,922
# entities	1,219,228	7,077,733
# distinct entities	75,549	192,138
# distinct triples	1,221,665	2,706,123
Avg. # contextual triples	21	19
Sparsity	98.70%	99.99%
Imbalanced ratio	48.39%	76.92%

sentiments from reviews to predict movie ratings. Ratings of all the movies are in the range of [0, 10].

2. **Amazon Movies and TV**. This dataset has been widely used to evaluate review rating prediction algorithms.[2] Ratings of all the movies are in the range of [0, 5].

We preprocessed the two datasets to ensure that each user had at least two reviews. The statistics of the two datasets after preprocessing are listed in Table 1. We randomly split each dataset into training, validation, and test sets with partition rates of 80%, 10%, and 10%, respectively, similar to [5, 6]. The hyperparameters were chosen from the best results on the validation set. All results are reported on the test set. We removed all reviews belonging to the test and validation sets, while building user and item representations on the basis of those reviews. For the CTR task, we transform ratings into implicit labels, where one and zero represent positive and negative feedback from a user to a movie, respectively. We followed the settings in previous works [16, 37] and set the threshold of a positive rating to 8 for the IMDb dataset and to 4 for the Amazon dataset, meaning that a result with an IMDb rating $\geq$ 8 or Amazon rating $\geq$ 4 is a positive label of 1, and otherwise is a negative label of 0.

5.2 Baselines

We compared the proposed KANN with the following state-of-the-art methods. The characteristics of all comparative models are listed in Table 2. Ratings and textual reviews are widely utilized as input features in current methods, and the KG is the auxiliary information introduced by knowledge-based RSs.

- Probabilistic matrix factorization (PMF). PMF models the latent factors for users and items while ignoring all the review texts [38].
- Generalized matrix factorization (GMF). This method extends matrix factorization (MF) to a nonlinear model that captures the interaction between users and items. It learns a latent embedding from a user (item) identifier to represent the corresponding user (item) [39].
- Multitask feature learning approach for KG-enhanced recommendation (MKR). This method maps item IDs to knowledge entities and performs a recommendation task and a KG embedding task at the same time [28].
- Deep cooperative neural networks (DeepCoNN). The DeepCoNN consists of two parallel neural networks that model user behaviors and item properties, respectively. It adopts a CNN layer and a shared layer to extract effective review features and model the interactions between users and items [16].
- Transformational neural networks (TransNets). It extends the DeepCoNN method by adding a transform layer. The layer transforms the review corresponding to the user-target item pair into an approximate representation for prediction [6].
- Multi-pointer co-attention networks for recommendation (MPCN). The MPCN is a review-by-review pointer-based model that uses hard attention to extract features from important reviews and then matches the important review representations in a word-by-word fashion [4].

5.3 Evaluation metrics

For the evaluation metric in the rating prediction task, we adopted the well-known root-mean-square error (RMSE), which is widely used in the rating prediction of recommendations. Given a ground-truth rating R_{ij} and a predicted rating $\widehat{R}_{ij}$ for an interaction between user i and movie j, the RMSE is calculated as

$$\mathrm{RMSE} = \sqrt{\frac{1}{|N|}\sum_{i,j\in N}\left(\widehat{r}_{ij} - r_{ij}\right)^2}, \tag{20}$$

where N indicates the set of samples.

For the CTR prediction task, we adopted accuracy (ACC) and area under ROC (AUC) as evaluation metrics. ACC can be formulated as

$$\mathrm{ACC} = \frac{1}{|N|}\sum_{i,j\in N}\mathbf{1}\left(\widehat{r}_{ij} = r_{ij}\right), \tag{21}$$

where $\mathbf{1}(\cdot)$ is an indicator function that outputs 1 if the condition is true and 0 otherwise. AUC can be formulated as

$$\mathrm{AUC} = \frac{1}{|N^+|}\frac{1}{|N^-|}\sum_{(i,j)\in N^+}\sum_{(i,j')\in N^-}\mathbf{1}\left(\widehat{r}_{ij} > \widehat{r}_{ij'}\right), \tag{22}$$

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

Table 2 Characteristics of baseline methods and KANN

Characteristics	PMF	GMF	MKR	DeepCoNN	TransNets	MPCN	KANN
Ratings	✓	✓	✓	✓	✓	–	–
Textual reviews	–	–	–	✓	✓	✓	✓
Knowledge graph	–	–	✓	–	–	–	✓
Explainable results	–	–	–	–	✓	✓	✓

where N^+ and N^- denote the set of positive samples and negative samples, respectively.

The lower the RMSE, the better the performance, and the higher the AUC and ACC, the better the performance.

5.4 Experimental setup

We optimized the weights of our model by Adam [40] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1.0 \times 10^{-9}$. The learning rate was customized with reference to another work [21]. The batch size was set to 128. The size of both knowledge embedding and latent embedding was set to 50 dimensions, giving $d = 100$ after concatenating the context entity with the entity itself. The hidden sizes of all layers were set to 1024. The number of spaces H was set to 4. We kept the length of reviews covering 90% of users and items, and the number of reviews covering 80% of users and items. We performed each experiment three times and reported the average performance on the test set when the validation RMSE was the lowest. Note that we adopted the early-stopping strategy to determine the convergence epochs, with epoch numbers of 35 and 40 for the IMDb and Amazon datasets, respectively. In addition, we recorded the data partition for each running time and used the same data partition on all benchmark methods.

For the PMF, the number of latent factors was 50, and the regularization parameter was 0.02.

For the GMF, we used the Adam optimizer, and the parameters were set to $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1.0 \times 10^{-9}$. The batch size was 128, the number of latent factors was 64, and the learning rate was 0.0001.

For the MKR, we set the number of high-level layers $K = 1$ and low-level layers $L = 2$. The dimensions of both user and entity embeddings were 8. The batch sizes of the Amazon and IMDb datasets were set to 64 and 32, respectively. The value of l_2 regularization was 1.0×10^{-6}, and the training intervals of the KGE task on Amazon and IMDb were 3 and 2, respectively. The ratio λ_1 of two learning rates was 0.04 and 0.1 on the IMDb and Amazon datasets, respectively.

DeepCoNN and TransNets used TextCNN [41] as the text processing method, and trained word embeddings by Gensim.[3] The length of the review document was set to 1000. The size of the word embeddings was 64, the number of neurons in the convolutional layer was 100, and the window size of the convolutional kernel was 3. The latent dimensions of the final user-item interaction layer were 10, the learning rate was 0.002, the dropout rate was 0.1, and the batch size was 128.

For the MPCN, the dropout rate was 0.2, the l_2 regularization was 1.0×10^{-6}, the number of pointers was 1, and the learning rate was 0.001. With reference to the settings reported in the MPCN [4], the length of the review document was set to 600 (20 reviews, 30 words per review). The size of word embeddings was 50, and the number of latent factors of the FM layer was 50.

5.5 Performance comparison

The average performance of all the comparative models on the test set for 2-core and 10-core is listed in Table 3. Here, the term "core" means the density of datasets; for example, 2-core means that each user in the dataset had at least two reviews [4, 6, 16].

First, we can see that the proposed KANN achieves the best performances among all the models on the IMDb dataset, and has the lowest RMSE for the rating prediction tasks and the highest ACC for the CTR tasks on the Amazon dataset. In particular, through observing the performances for the rating prediction tasks, we can see that KANN improves over the best baselines by 1.39% and 2.64% on the IMDb and Amazon datasets, respectively. Through checking the performances for the CTR tasks, we can see that the corresponding results are almost the same as those of the other tested methods. Although the improvements in our proposed model over some state-of-the-art models may not seem relatively significant, it is clear that the performance of our model achieves the state-of-the-art level. The above outcomes empirically demonstrate the relatively high effectiveness of our proposed model.

We can also see that the models extracting features from reviews (DeepConn, TransNets, MPCN, and KANN) outperform the methods that do not. This clearly indicates that

[3] https://radimrehurek.com/gensim/.

Table 3 Performance comparison with baselines

Dataset	#Cores	Metric	Methods						
			PMF	GMF	MKR	DeepCoNN	TransNets	MPCN	KANN
IMDb	2	RMSE	1.916	1.865	1.838	1.815	1.814	1.862	**1.782**
		AUC	0.805	0.775	0.763	0.826	0.827	0.814	**0.827**
		ACC	0.726	0.705	0.692	0.729	0.726	0.727	**0.750**
	10	RMSE	1.758	1.771	1.802	1.776	1.766	1.806	**1.74**
		AUC	0.807	0.779	0.763	0.828	0.832	0.827	**0.835**
		ACC	0.734	0.709	0.688	0.727	0.729	0.731	**0.751**
Amazon	2	RMSE	1.215	1.170	1.103	1.037	1.037	1.043	**1.012**
		AUC	0.735	0.542	0.708	0.773	**0.776**	0.771	0.753
		ACC	0.679	0.769	0.771	0.787	0.792	0.780	**0.793**
	10	RMSE	1.097	1.148	1.060	1.016	1.021	1.009	**0.980**
		AUC	0.772	0.572	0.752	0.785	**0.792**	**0.792**	0.780
		ACC	0.736	0.775	0.763	0.775	0.777	0.772	**0.786**

The best results are highlighted in bold

RSs leveraging textual reviews as additional information can result in significantly improved performances.

5.6 Impact of different sparsities

The cold-start [42] problem is a prevalent research topic in recommender systems. It has been proven that review information can effectively alleviate the cold-start problem, especially when the number of reviews is relatively small [16, 43]. In order to examine the effectiveness of our model on different sparsities, we conducted evaluations by varying the minimal number of reviews K for each user from 2 to 10 on the two datasets. We show the RMSE results for all the comparative models on different K in Fig. 3.

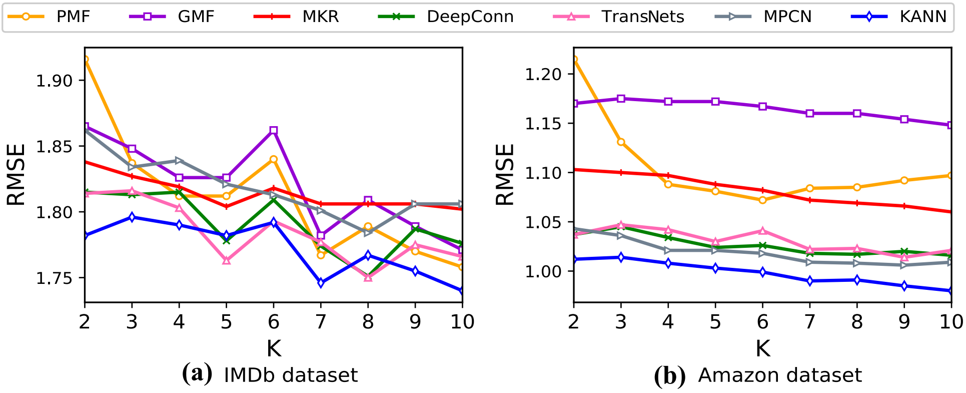
We can see here that most methods perform better with the increase of K. On the Amazon dataset, our proposed KANN outperforms all other models under different values of K. Even on the IMDb dataset, KANN achieves the best performance except in the cases of $K = 5$ and 8. Moreover, the RMSEs of KANN under a relatively small number of reviews ($K = 2, 3, 4$) were lower than those of the other models, which indicates that our model can alleviate the cold-start problem more effectively than the baselines. In general, we can observe that, while all the models benefited from the increase of K, the corresponding improvements in performances were more significant for KANN than for the others, which demonstrates that our proposed KANN can leverage review information more effectively than the baselines.

6 Analysis of KANN

In this section, we conduct ablation studies to verify the effectiveness of our model (Sect. 6.1), analyze the effect of the number of projected knowledge semantic spaces (Sect. 6.2), provide a visual explanation of the effectiveness of multi-space attention (Sect. 6.3), and share our insights on KANN in terms of the attention mechanism and the user-movie interaction through a case study (Sect. 6.4). Lastly, we share the knowledge explainability results and compare the explainability with representative baseline models (Sects. 6.5 and 6.6, respectively).

6.1 Ablation study

To investigate the effectiveness of three key parts in our model, namely the inner-attention mechanism, the outer-attention mechanism, and the knowledge embeddings learned by TransE, we conducted ablation studies on two datasets. Specifically, we first disabled the inner-attention mechanism of KANN by removing the inner-attention operations, termed KANN-inner, and then disabled the outer-attention mechanism by removing the outer-attention operations, termed KANN-outer. Lastly, we replaced the knowledge embeddings learned by TransE [34] with embeddings generated by a fully connected layer on the KG, termed KANN-TransE. The results are shown in Table 4, where we can see that disabling any of the three key parts degrades the performance of KANN. KANN performs better than KANN-inner and KANN-outer, which means that our low-order interaction learning between users and movies is playing a non-negligible role in recommendation. KANN-TransE underperforms the other two ablation methods on the Amazon dataset, but performs



(a) IMDb dataset **(b)** Amazon dataset

Fig. 3 Comparison of RMSE results between KANN and baselines on different sparsities

Table 4 RMSE results of ablation study

Dataset	KANN-TransE	KANN-inner	KANN-outer	KANN
IMDb	1.795	1.802	1.789	**1.782**
Amazon	1.036	1.013	1.014	**1.012**

better than KANN-inner on the IMDb dataset. The reason maybe that the KG of the Amazon dataset is much larger than that of the IMDb dataset, and a simple fully-connected layer cannot completely extract the representations of entities in the KG. The above results demonstrate the necessity of all three key parts in KANN.

6.2 Effect of number of spaces

In the proposed model, we project the knowledge embedding inputs into H spaces to generate diverse expressions of knowledge semantic features. We compare the RMSEs and the running time of each epoch under different numbers of knowledge semantic spaces in Table 5, where running time is measured in seconds. Since the embedding size d was set to 100 and needs to be evenly divisible by space value, we vary the value of H in the range of [1, 2, 4, 5, 10, 20, 25] and observe the corresponding RMSEs on two datasets. The results show that the increase in the number of spaces H was beneficial for yielding a lower RMSE on both datasets until H reached 20. We can also see that as the number of spaces increased, the running time increased as well. Therefore, after considering the calculation time and performance results, we used the experimental results of $H = 4$ in Sect. 5.

6.3 Multi-space attention visualization and explanation

In this subsection, we visualize the interaction of a user and a movie based on multiple spaces. To illustrate the multi-space attention of our proposed model, we randomly selected one pair of a user (user A) and a movie (movie B) from the test set of the Amazon dataset. For ease of illustration, we trained KANN with $H = 4$ and only visualized some of the entities. Here, we show the first 35 of 60 user entities and the first 35 of 210 movie entities. Figure 4 shows the affinity matrix varying H from 1 to 4, where the x-axis and y-axis labels represent indexes of the user entities and the movie entities, respectively. Each cell in Fig. 4 is a weight value from a movie entity to a user entity. The goal was to select entities related to user A from the movie reviews. Similarly, Fig. 5 shows the user review entities important to movie B, where the x-axis labels are the movie entity indexes, the y-axis labels are the user entity indexes, and each cell is a weight value from a user entity to a movie entity. The goal was to select entities related to movie B from the user reviews. In both figures, the brighter the color is, the higher the weight value, where higher weights indicate entities more important to the user/movie.

Visualizing the interaction weights clearly demonstrates the necessity of modeling multiple spaces. Since the important entities in each space are different, using

Table 5 RMSE results for two datasets with different numbers of knowledge semantic space

#spaces		1	2	4	5	10	20	25
IMDb	RMSE	1.778	1.781	1.782	1.776	1.754	**1.717**	1.722
	Time (s)	203	260	360	404	725	1400	1638
Amazon	RMSE	1.015	1.014	1.012	1.013	1.011	**1.010**	1.012
	Time (s)	622	656	708	736	856	1151	1305

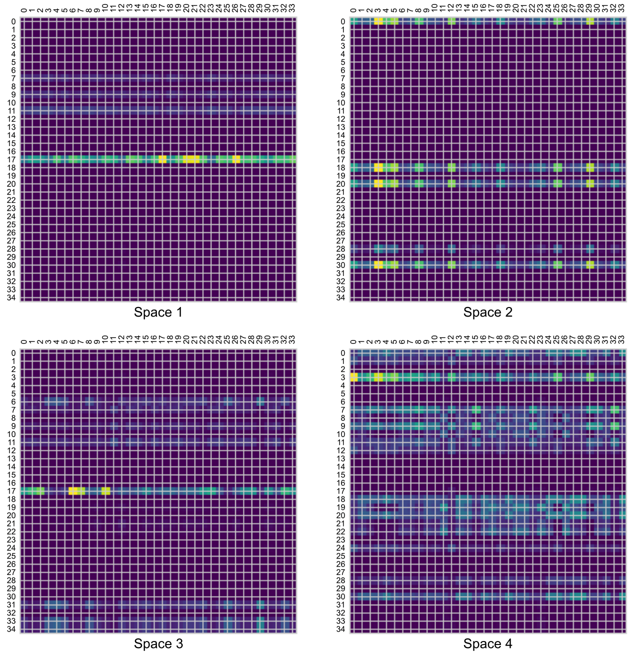


Fig. 4 Visualization of entity-level attention for movie entities related to user A

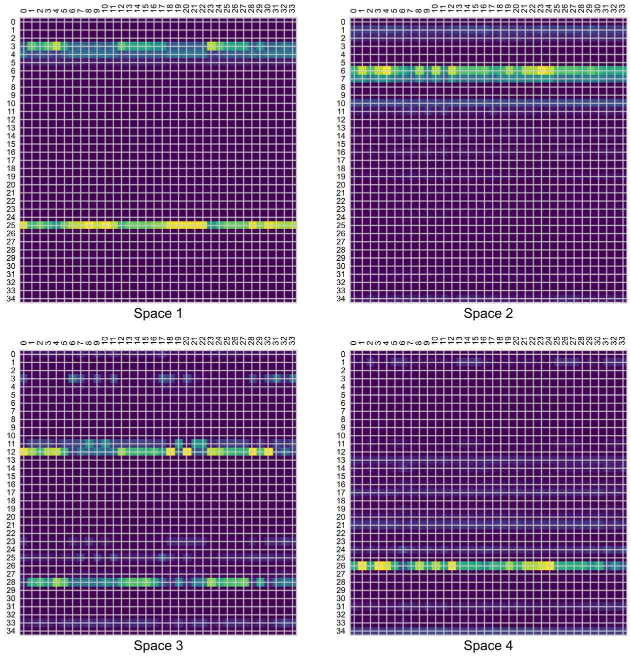


Fig. 5 Visualization of entity-level attention for user entities related to movie B

multiple spaces is beneficial to effectively extract features and improve performances.

6.4 Case study

First, we utilize the randomly selected user-movie pair (A, B) in Sect. 6.2 to provide detailed information of the important review entities and the effectiveness of our knowledge entity distillation operation. Then, we show the relations between review entities in a KG and the interactions between user A and movie B.

The important knowledge entities of user A and movie B can be understood by analyzing the visualization figures and corresponding entity information. We list the selected entities corresponding to user A and movie B in Tables 6 and 7, respectively. In both tables, one row represents one entity, including its index, name, description, and the review where it is located, and each index corresponds to a highlighted row in Figs. 4 and 5. The words in bold from reviews are the mentions corresponding to the selected knowledge entities. The indexes in each space are depicted in descending order of attention weight. We can obtain important movie entities related to user A by observing the highlighted rows in Fig. 4. The indexes of the important entities to user A in Fig. 4 are $\{17, 11, 7, 9\}$, $\{0, 18, 20, 30, 28\}$, $\{17, 33, 34, 31, 6, 11\}$, and $\{3, 7, 9, 0, 18, 20, 30, 19, 22, 11\}$ from spaces 1 to 4. Similarly, the indexes of the important entities to movie B are $\{25, 3, 4, 5\}$, $\{6, 7, 10, 1, 2\}$, $\{12, 28, 11, 33, 3\}$, and $\{26, 34, 21, 24, 17, 13\}$ from spaces 1 to 4 in Fig. 5.

In Table 6, the important entities to user A selected from movie B include multiple same entities. This is because we removed duplicated entities in the same review but retained repeated entities across different reviews. In this way, entities that are mentioned by different users can show their importance when they appear multiple times. The indexes of 0, 18, 20, and 30 in the second space of Table 6 point to the same entity, and movie B also corresponds to this entity. The genre of movie B is "drama" with index 6, and the movie is also an "independent film" with the indexes 33 and 34. Combined with the information in Fig. 5, we can determine that the index with the brightest color is 25 in Table 7, which is the movie genre mentioned by user A in the reviews, and this movie genre is important to movie B. We can also see several war-related entities in spaces 2 and 3, since the movie "Yankee Doodle Dandy" that user A watched with index 33 is a war film. We also selected several actors that the user mentioned in the reviews in space 4, which are important for movie B. Overall, most entities in the two tables are meaningful and reasonable according to human intuition, which demonstrates the effectiveness of our knowledge distillation.

We can clarify the interactions between user A and movie B by analyzing important entities and the relationships between them. We draw a graph based on all the

Table 6 Detailed information on selected movie entities corresponding to user A

Space	Index	Name	Description	Review
1	17	All	American band	WOW! What a gem! This movie is quiet and unassuming, yet very powerful. **ALL** of the performances (Dinklage, Cannavale, Clarkson) are spot on.
	11	Joe Matt	Autobiographical cartoonist	**Spent** the whole movie waiting for the funny that was advertised...
	7,9	Carnival	Festive season which occurs immediately before Lent	Clarkson and **Cannavale** are delightful as Fin’s new friends... Great acting.. Loved Bobby **Cannavale’s** character...
2	0,18,20,30	The Station Agent	2003 film directed by Tom McCarthy	**THE STATION AGENT** is Finbar McBride and he is played by... “**The Station Agent**” is a quirky movie... “**The Station Agent**” is an offbeat and engaging tale... Wonderful story of 4 people who are drawn by the charisma of **The Station Agent**...
	28	Raven Goodwin	American actress	The young Michelle Williams (as a librarian) and **Raven Goodwin** as Cleo...
3	17	All	American band	WOW! What a gem! This movie is quiet and unassuming, yet very powerful. **ALL** of the performances (Dinklage, Cannavale, Clarkson) are spot on.
	33,34	Independent film	Film production mostly or completely done outside of the major film studio system	Good **Indie movie** with some of my favorite actors......Patricia Clarkson... Usually I enjoy **Independent films**, but this one......
	31	Horror film	Film genre	**Horror**, especially, is the genre in which I will watch almost anything...
	6	Drama	Artwork intended for performance, formal type of literature	In totality, this is a motion picture that blends uplifting theatrics and climactic **dramatics** together quite flawlessly...
	11	Joe Matt	Autobiographical cartoonist	**Spent** the whole movie waiting for the funny that was advertised...
4	3	Gorgeous	1999 Hong Kong action Romantic comedy film directed by Vincent Kok	Joe (Bobby Cannavale) runs his ailing dad’s hot dog “emporium **Gorgeous** Frank’s, and Olivia, a recently divorced, forty-something writer (Patricia Clarkson)...
	7,9	Carnival	Festive season which occurs immediately before Lent	Clarkson and **Cannavale** are delightful as Fin’s new friends... Great acting.. Loved Bobby **Cannavale’s** character...
	0,18,20,30	The Station Agent	2003 film directed by Tom McCarthy	**THE STATION AGENT** is Finbar McBride and he is played by... “**The Station Agent**” is a quirky movie... “**The Station Agent**” is an offbeat and engaging tale... Wonderful story of 4 people who are drawn by the charisma of **The Station Agent**...
	19	Toronto	Capital city of the province of Ontario, Canada	I first saw this film when on an airplane trip from **Toronto** to Los Angeles...
	22	Manhattan	Borough of New York City	The two most prominent are Joe, a boisterous Cuban-American from **Manhattan** taking care of his ailing father’s hot dog stand, and Olivia...
	11	Joe Matt	Autobiographical cartoonist	**Spent** the whole movie waiting for the funny that was advertised...

important entities (some of which are listed in Tables 6 and 7) of user A and movie B, as shown in Fig. 6.

We analyze the effectiveness of our model from two aspects based on the analysis of Fig. 6. First, the extracted review entities are not limited to the current movie B but also include other movie information such as attributes or subjects related to movie B that cannot be obtained from the movie database. In this case, the movie “Lenny” in the movie graph and the actors “Gene Autry” and “Fred Astaire” in the user graph are not directly related to the current movie (“The Station Agent”). However, they can be captured by our model from reviews on the basis of their KG relationships with the movie. Therefore, the knowledge

Table 7 Detailed information on selected user entities corresponding to movie B

Space	Index	Name	Description	Review
1	25	Western	Multimedia genre of stories set primarily in the American Old West	I'm a huge fan of **Westerns**, from Gene Autry to John Wayne to Clint Eastwood...
	3	Foam	Form of matter	She shared her favors and a bottle of Angel **Foam** with the town's un-uniformed cop, Griff...
	4	Pete Conway	American baseball player and coach	What isn't ignored, unfortunately, is the story of rancorous PFC **Pete Conway**, the Agar character...
	5	Agar	Thickening agent used in microbiology and food	**Agar** in the movie, which goes to show you that great actors don't necessarily make great casting directors...
2	6	Conway	City in Arkansas	**Conway** woos a USO women, played by Allison Bromley...
	7	Bromley	Large suburban district of South East London	Conway woos a USO woman, played by Allison **Bromley**...
	10	United States Marine Corps	Branch of the United States Armed Forces	In any event, war breaks out and the under-aged Murphy is turned down by both the **Marines** and the Navy...
	1	Bon-Bon	Short story by Edgar Allan Poe	At least I didn't mention the **Bon-Bon** Girls in the town across the river or the room Kelly rented from the woman...
	2	The Naked Kiss	1964 film by Samuel Fuller	Two-thirds of the way through **THE NAKED KISS** a wardful of pediatric orthopedic patients...
3	12	Pacific Ocean theatre of World War II	The naval and island campaigns in the Central Pacific, North Pacific and South Central Pacific	Hollywood,World War II,**Pacific Theater**,Wake Island,Bataan...
	28	Clint Eastwood	American actor and film director	I'm a huge fan of Westerns, from Gene Autry to John Wayne to **Clint Eastwood** and all stops...
	11	Sergeant	Military rank	Wayne plays Marine **Sergeant** John Stryker, a tough-as-nails type who has to whip his squad into shape during a rest...
	33	Yankee Doodle Dandy	1942 film by Michael Curtiz	For anyone (like me) who's used to Cagney as a gun-toting gangster **YANKEE DOODLE DANDY** will come as something of a revelation...
	3	Foam	Form of matter	She shared her favors and a bottle of Angel **Foam** with the town's un-uniformed cop, Griff...
4	26	Gene Autry	American actor and singer	I'm a huge fan of Westerns, from **Gene Autry** to John Wayne to Clint Eastwood and all stops...
	34	Fred Astaire	American dancer, singer, actor, choreographer, and television presenter	Jimmy can sing (well, he sings no worse than **Fred Astaire**, anyway) and he can dance...
	21	Slim Pickens	American actor and rodeo performer	Sam recovers, though, Uncle Beck organizes a chase party (including Royal Dano, Dewey Martin, and **Slim Pickens**)...
	24	Pete Rose	American baseball player	McCABE MRS. MILLER killed the genre–That's kind of like saying **Pete Rose** destroyed baseball...
	17	Marta Kristen	Norwegian actress	Sam can track just about anything, which comes in handy when he and Arliss go off chasing a egg-stealing bobcat and along with Travis and young Lisbeth Searcy (**Marta Kristen**,)...
	13	John Agar	American actor	Joanne Dru plays the niece, while **John Agar** and Harry Carey Jr. play two young lieutenants...

entities of reviews are complementary features that cannot be obtained from other content.

Second, to clarify the interactions between the review entities of the graph shown in Fig. 6, we focus on the relations of the figure in this part. For the inner relations of movie entities, "genre", "distributed by", and "cast member" are the three main kinds of relation starting from two movie entities. For the inner relationships of user entities, the main relations are "genre", "director", "distributed by", "cast member", and "subject" starting from movie entities. For the outer interactions between user A and movie B, "genre" and "distributed by" are the outer relations between the user and the movie. On the basis of this analysis, we can see that most of the relations are directly related to movies. The most important finding here is that our model can not only model the inner relations

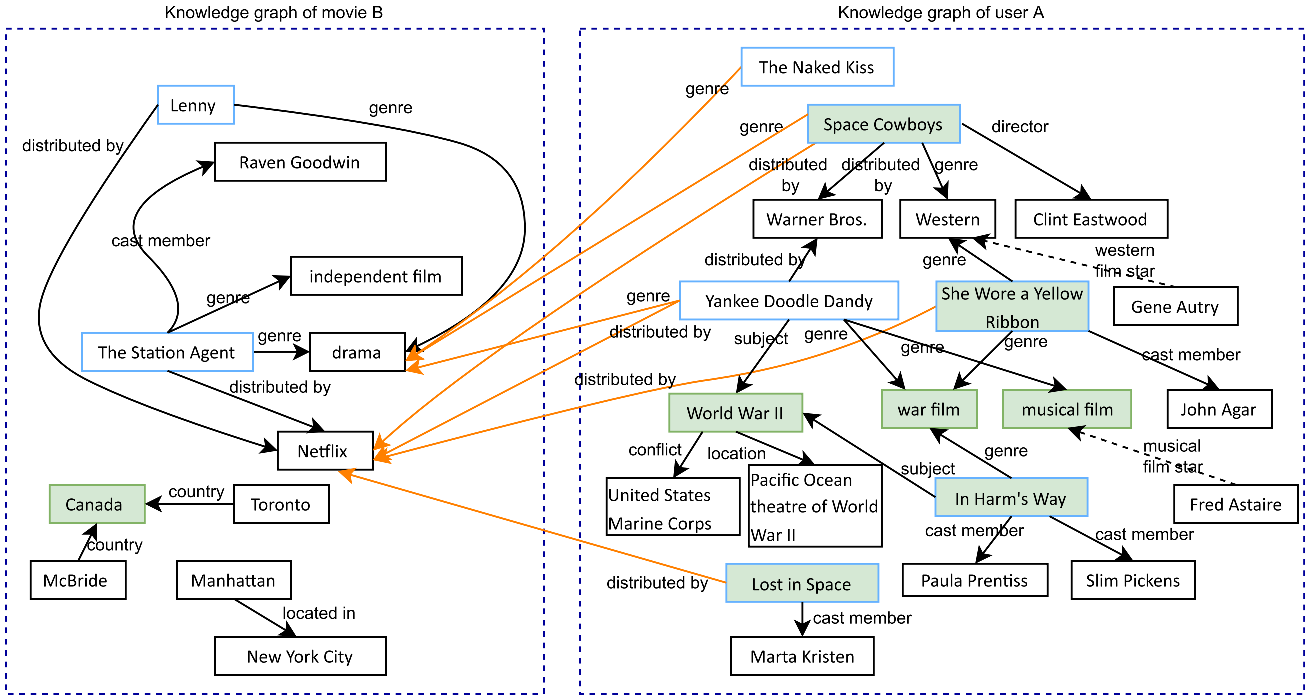


Fig. 6 KG between user A and movie B. Each box represents an entity, and the word in it is the name of the entity. The boxes without filling color are entities that are selected by KANN from review entities, where the blue-box nodes are movie entities. The blue-box nodes with green filling color are movies that have been watched by user A. Green-box nodes are introduced entities to connect the other entities. The arrows represent the relations between entities, where the orange arrows represent the outer relations between the user and the movie. The dashed lines represent the two-hop distance relation in the user KG

between user/movie review entities but also learn the outer interactions between a user and a movie. It is worth noting that our model is not simply exploiting the connections between entities but is modeling the user preferences and movie properties based on the inner and outer interaction learning.

6.5 Statistics of user preferences

To determine the overall distribution of user preferences and further reveal the explainability of our work, we count the important entity categories of our two test datasets from 13 aspects that most users pay attention to. In this way, we can find out what kinds of knowledge entities are typically found by learning the interactions between users and movies. In particular, the statistical categories can provide

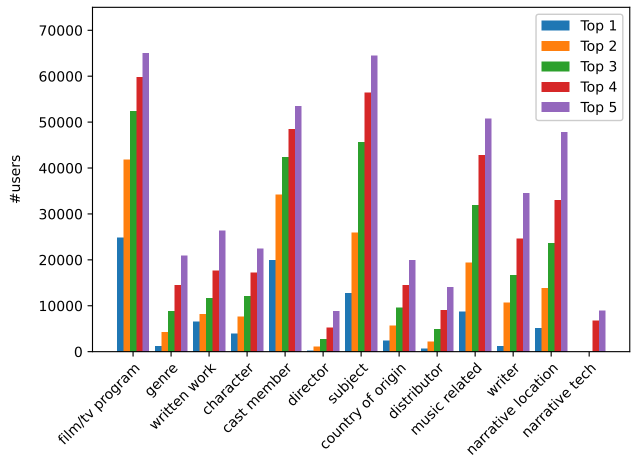


Fig. 7 Distribution of user preferences in Amazon dataset

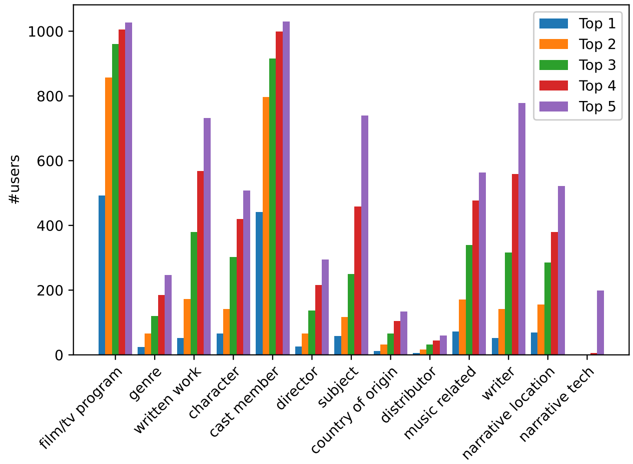


Fig. 8 Distribution of user preferences in IMDb dataset

insights into user behaviors in the movie domain and help improve the service quality of movie recommendations.

To clarify user preferences, we first select important entities whose weights are greater than the average weight, and obtain the categories corresponding to the selected entities. Then, we count the occurrences of the entity categories corresponding to each user. Lastly, we order the categories of the selected entities from the top 1 to the top 5 that users pay attention to by ranking the occurrence numbers of entity categories in each user. The statistic results are accumulated from the top 1 to the top 5. Distributions of user preferences corresponding to the Amazon and the IMDb datasets are shown in Figs. 7 and 8, respectively, where the x-axis is the entity category and the y-axis is the number of users. We can see that the most relevant entity category for users in both datasets is "film/tv program", followed by "cast member". The main difference between the two datasets is that users in the Amazon dataset pay more attention to "subject" than those in the IMDb dataset. We also find that users are very interested in the "music related" category, "writer", and "narrative location" of movies in both datasets. The "music related" category includes musical works, composers, singers, and so on. To our surprise, the results suggest that users do not pay much attention to directors but rather focus on the "character" and the "film/tv program".

6.6 Comparison of explainability with current works

Next, to investigate the advantages of our model in terms of explainability, we compared KANN with the existing representative models discussed in Sect. 2.

In Table 8, we use an example to illustrate different expression forms for explainability corresponding to the current review-based works and KANN. The explanations of current review-based models can be divided into three levels: the review level, the word level, and the knowledge level. TransNets [6] and NARRE [5] use word embedding to express reviews, and provide review-level explanations. The explainable results of MPCN [4] and CARL [44] can be obtained from the word level. Our proposed model can provide knowledge-level explanations. Since the structures of selected triples from the KG are fixed, we can generate more understandable explanations for end users based on template sentences, e.g., *we recommend ______(a movie) to you, because the ______(the relation of the triple) of ______(the subject of the triple) is ______(the object of the triple)*. In Table 8, we can observe that our knowledge-level explanations are easier to understand than review-level and word-level explanations. This is because the review-based models can only highlight the words or reviews with a high weight value. In most cases, users are required to read the entire review or the context sentences to understand the scattered highlighted words.

Table 9 shows the advantages of KANN in terms of explainability from four aspects in comparison with current

Table 8 Comparison of explanations of review-based models using real reviews from Amazon dataset

Recommended movie	Review-level and word-level explanations		Knowledge-level explanations
	TransNets/NARRE	MPCN/CARL	KANN
THE STATION AGENT	The Station agent is the story of Finbar McBride (indie favorite Peter Dinklage, of Safe Men and Living in Oblivion fame), a train enthusiast from north Jersey. This whole cast is simply tremendous, but it is Clarkson who truly shines. She always brings a touch of greatness to everything she touches. Goodwin proves herself a capable enough actress at playing herself.	Writer/director Tom McCarthy has made **a most unique** character study in 2003's **"The Station Agent"**. On the sidelines, there are **nice, affecting** turns by a **better-than-expected** Michelle Williams as Emily and **Raven Goodwin** as Cleo. Unfortunately there are also some **melodramatic flourishes**, such as Fin's **drunken assertion** at the bar, his hurtful rejection by **Olivia**, and the appearance of Emily's white-trash boyfriend, that don't seem to serve much purpose **beyond** confirming the fact that Fin needs love and friends.	The station Agent→cast member→Raven Goodwin (We recommend "The Station Agent" to you, because the cast member of "The Station Agent" is Raven Goodwin.) The Station Agent→genre→drama, Yankee Doodle Dandy→genre→drama (We recommend "The Station Agent" to you, because the genre of "The Station Agent" is drama, and the genre of "Yankee Doodle Dandy" you watched before is drama too.) The Station Agent→distributed by→Netflix, Yankee Doodle Dandy→distributed by→Netflix (......)

Table 9 Comparison with current works in terms of explainability

	TransNets	NARRE	MPCN	CARL	HeteRec	DKER	RuleRec	KANN
Does not require expert manual effort	✓	✓	✓	✓	–	–	✓	✓
Review-based explanations	✓	✓	✓	✓	–	–	–	✓
Knowledge-level explanations	–	–	–	–	✓	✓	✓	✓
User preference analysis	–	–	–	–	–	–	–	✓

representative explainable RSs. HeteRec [45] and DKER [29] are knowledge-based methods that require expert manual definitions of various useful meta-paths to provide an explainable result. Although RuleRec [30] induces rules from item associations automatically, it cannot provide analyses about user preferences. In contrast to the above works, our model can distill knowledge information from movie reviews and model low-order interactions between users and movies. These functions enable it to provide highly explainable recommendations at the knowledge level without manual effort, and to supply detailed analyses of user preferences based on user-movie interactions.

7 Conclusion

In this work, we proposed KANN, a model to deal with movie recommendation tasks with high explainability. KANN utilizes the attention mechanism leveraged in multiple knowledge semantic spaces to model understandable interactions between users and movies at the knowledge level. Our experimental results show that KANN outperforms many state-of-the-art methods on the IMDb and Amazon datasets for predicting ratings and CTR tasks. By exploring the potential connections between users and movies, the proposed KANN can provide highly convincing and accurate recommendations to users. Our analyses of the distribution of user preferences further strengthens the explainability of our model.

As future work, we will assess the persuasiveness of explanations given by our model through implementing a crowdsourcing evaluation. We will also extend KANN to model auxiliary information such as temporal information. Moreover, we are interested in exploring more effective review features from multiple perspectives, such as the word and character levels.

Acknowledgements This work was partly supported by JSPS KAKENHI Grant Numbers 18H03242, 18H03342, and 19H01138.

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

References

1. He X, Chen T, Kan M-Y, Chen X (2015) Trirank: review-aware explainable recommendation by modeling aspects. In: Proceedings of the 24th ACM international on conference on information and knowledge management. ACM, pp 1661–1670
2. Zhang Y, Lai G, Zhang M, Zhang Y, Liu Y, Ma S (2014) Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 83–92
3. Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv (CSUR) 52(1):5
4. Tay Y, Luu AT, Hui SC (2018) Multi-pointer co-attention networks for recommendation. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 2309–2318
5. Chen C, Zhang M, Liu Y, Ma S (2018) Neural attentional rating regression with review-level explanations. In: Proceedings of the 2018 world wide web conference. International World Wide Web Conferences Steering Committee, pp 1583–1592
6. Catherine R, Cohen W (2017) Transnets: learning to transform for recommendation. In: Proceedings of the eleventh ACM conference on recommender systems. ACM, pp 288–296
7. Seo S, Huang J, Yang H, Liu Y (2017) Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: Proceedings of the eleventh ACM conference on recommender systems. ACM, pp 297–305
8. Cheng Z, Ding Y, He X, Zhu L, Song X, Kankanhalli MS (2018) A 3ncf: an adaptive aspect attention model for rating prediction. In: IJCAI, pp 3748–3754

9. Wang H, Zhang F, Xie X, Guo M (2018) Dkn: Deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 world wide web conference. International World Wide Web Conferences Steering Committee, pp 1835–1844
10. Yang B, Mitchell T (2019) Leveraging knowledge bases in lstms for improving machine reading. arXiv:1902.09091
11. Wang J, Wang Z, Zhang D, Yan J (2017) Combining knowledge with deep convolutional neural networks for short text classification. In: IJCAI, pp 2915–2921
12. Tan Y, Zhang M, Liu Y, Ma S (2016) Rating-boosted latent topics: understanding users and items with ratings and reviews. IJCAI 16:2640–2646
13. Liu H, Wang Y, Peng Q, Wu F, Gan L, Pan L, Jiao P (2020) Hybrid neural recommendation with joint deep representation learning of ratings and reviews. Neurocomputing 374:77–85
14. Diao Q, Qiu M, Wu C-Y, Smola AJ, Jiang J, Wang C (2014) Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 193–202
15. Peña FJ, O'Reilly-Morgan D, Tragos EZ, Hurley N, Duriakova E, Smyth B, Lawlor A (2020) Combining rating and review data by initializing latent factor models with topic models for top-n recommendation. In: Fourteenth ACM conference on recommender systems, pp 438–443
16. Zheng L, Noroozi V, Yu P.S (2017) Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the tenth ACM international conference on web search and data mining. ACM, pp 425–434
17. Rendle S (2010) Factorization machines. In: 2010 IEEE international conference on data mining. IEEE, pp 995–1000
18. Almahairi A, Kastner K, Cho K, Courville A (2015) Learning distributed representations from reviews for collaborative filtering. In: Proceedings of the 9th ACM conference on recommender systems. ACM, pp 147–154
19. Ren Z, Liang S, Li P, Wang S, de Rijke M (2017) Social collaborative viewpoint regression with explainable recommendations. In: Proceedings of the tenth ACM international conference on web search and data mining. ACM, pp 485–494
20. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv:1409.0473
21. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A.N, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
22. Zhang F, Yuan NJ, Lian D, Xie X, Ma W-Y (2016) Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 353–362
23. Wang H, Zhang F, Wang J, Zhao M, Li W, Xie X, Guo M (2018) Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM international conference on information and knowledge management. ACM, pp 417–426
24. Tai C-Y, Huang L-Y, Huang C-K, Ku L-W (2021) User-centric path reasoning towards explainable recommendation. In: Proceedings of the 44th International iCM SIGIR conference on research and development in information retrieval, pp 879–889
25. Liu Y, Miyazaki J, Chang Q (2022) Knowledge graph enhanced multi-task learning between reviews and ratings for movie recommendation. In: Proceedings of the 37th ACM/SIGAPP symposium on applied computing, pp 1882–1889
26. Cheekula SK, Kapanipathi P, Doran D, Jain P, Sheth AP (2015) Entity recommendations using hierarchical knowledge bases. CEUR Workshop Proc 1365:2015
27. Zhu Q, Zhou X, Wu J, Tan J, Guo L (2020) A knowledge-aware attentional reasoning network for recommendation. Proc AAAI Conf Artif Intell 34:6999–7006
28. Wang H, Zhang F, Zhao M, Li W, Xie X, Guo M (2019) Multi-task feature learning for knowledge graph enhanced recommendation. In: The world wide web conference, pp 2000–2010
29. Zhang Y, Xu X, Zhou H, Zhang Y (2020) Distilling structured knowledge into embeddings for explainable and accurate recommendation. In: Proceedings of the 13th international conference on web search and data mining, pp 735–743
30. Ma W, Zhang M, Cao Y, Jin W, Wang C, Liu Y, Ma S, Ren X (2019) Jointly learning explainable rules for recommendation with knowledge graph. In: The world wide web conference, pp 1210–1221
31. Wang X, He X, Cao Y, Liu M, Chua T-S (2019) Kgat: Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, pp 950–958
32. Ratinov L, Roth D, Downey D, Anderson M (2011) Local and global algorithms for disambiguation to wikipedia. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, vol 1. Association for Computational Linguistics, pp 1375–1384
33. Cheng X, Roth D (2013) Relational inference for wikification. In: Proceedings of the 2013 conference on empirical methods in natural language processing, pp 1787–1796
34. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems, pp 2787–2795
35. Han X, Cao S, Xin L, Lin Y, Liu Z, Sun M, Li J (2018) Openke: an open toolkit for knowledge embedding. In: Proceedings of EMNLP
36. Zhou C, Bai J, Song J, Liu X, Zhao Z, Chen X, Gao J (2018) Atrank: an attention-based user behavior modeling framework for recommendation. In: Thirty-second AAAI conference on artificial intelligence, pp 4564–4571
37. Kotzias D, Denil M, De Freitas N, Smyth P (2015) From group to individual labels using deep features. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 597–606
38. Mnih A, Salakhutdinov RR (2008) Probabilistic matrix factorization. In: Advances in neural information processing systems, pp 1257–1264
39. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web. International World Wide Web Conferences Steering Committee, pp 173–182
40. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
41. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, pp 1746–1751. https://doi.org/10.3115/v1/D14-1181. https://www.aclweb.org/anthology/D14-1181
42. Schein AI, Popescul A, Ungar LH, Pennock DM (2002) Methods and metrics for cold-start recommendations. In: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, pp 253–260
43. McAuley J, Leskovec J (2013) Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM conference on recommender systems, pp 165–172

44. Wu L, Quan C, Li C, Wang Q, Zheng B, Luo X (2019) A context-aware user-item representation learning for item recommendation. ACM Trans Inf Syst (TOIS) 37(2):1–29

45. Yu X, Ren X, Sun Y, Gu Q, Sturt B, Khandelwal U, Norick B, Han J (2014) Personalized entity recommendation: a heterogeneous information network approach. In: Proceedings of the 7th ACM international conference on web search and data mining, pp 283–292

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.