entropy

MDPI

Article

Exploiting Dual-Attention Networks for Explainable Recommendation in Heterogeneous Information Networks

Xianglin Zuo [1,2], Tianhao Jia [1,2], Xin He [2,3,*], Bo Yang [1,2] and Ying Wang [1,2,*]

[1] College of Computer Science and Technology, Jilin University, Quanjin Street, Changchun 130012, China
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Quanjin Street, Changchun 130012, China
[3] School of Artificial Intelligence (SAI), Jilin University, Quanjin Street, Changchun 130012, China
* Correspondence: hexin20@mails.jlu.edu.cn (X.H.); wangying2010@jlu.edu.cn (Y.W.)

Abstract: The aim of explainable recommendation is not only to provide recommended items to users, but also to make users aware of why these items are recommended. Traditional recommendation methods infer user preferences for items using user–item rating information. However, the expressive power of latent representations of users and items is relatively limited due to the sparseness of the user–item rating matrix. Heterogeneous information networks (HIN) provide contextual information for improving recommendation performance and interpreting the interactions between users and items. However, due to the heterogeneity and complexity of context information in HIN, it is still a challenge to integrate this contextual information into explainable recommendation systems effectively. In this paper, we propose a novel framework—the dual-attention networks for explainable recommendation (DANER) in HINs. We first used multiple meta-paths to capture high-order semantic relations between users and items in HIN for generating similarity matrices, and then utilized matrix decomposition on similarity matrices to obtain low-dimensional sparse representations of users and items. Secondly, we introduced two-level attention networks, namely a local attention network and a global attention network, to integrate the representations of users and items from different meta-paths for obtaining high-quality representations. Finally, we use a standard multi-layer perceptron to model the interactions between users and items, which predict users' ratings of items. Furthermore, the dual-attention mechanism also contributes to identifying critical meta-paths to generate relevant explanations for users. Comprehensive experiments on two real-world datasets demonstrate the effectiveness of DANER on recommendation performance as compared with the state-of-the-art methods. A case study illustrates the interpretability of DANER.

Keywords: heterogeneous information networks; dual attention mechanism; rating prediction; meta-path

Citation: Zuo, X.; Jia, T.; He, X.; Yang, B.; Wang, Y. Exploiting Dual-Attention Networks for Explainable Recommendation in Heterogeneous Information Networks. *Entropy* **2022**, *24*, 1718. https://doi.org/10.3390/e24121718

Academic Editors: Yilun Shang and Alexandre G. Evsukoff

Received: 30 September 2022
Accepted: 15 November 2022
Published: 24 November 2022

1. Introduction

Recommendation systems have been widely used in various online services, such as search engines, e-commerce, online news, and social media sites, and have become one of the most powerful ways to solve the problem of information overload [1,2]. However, a large number of recommendation methods are still black-boxes that do not provide explanations for users. In recent years, explainable recommendation has attracted increasing attention in the academic and industrial communities. Explainable recommendation systems not only unveil the recommendation process, but also help to improve the effectiveness, persuasiveness and satisfaction of the recommendations.

Traditional recommendation methods, e.g., matrix factorization, mainly infer the preferences of users for items by using implicit or explicit user–item interaction data [3]. The key to generating accurate recommendation results is to obtain the representations of users and items with rich expressive power, while these traditional methods suffer from

the sparseness of interaction data [4,5]. A common idea which can solve the problem of data sparseness is to introduce some auxiliary information into the recommendation system. Auxiliary information can make up for the sparseness or lack of interaction data, enrich preferences of users and features of items and enhance the performance of the recommendation system effectively [6]. What is more, traditional recommendation methods only provide some simple explanations, such as *"Customers Who Bought This Item Also Bought…"*, with which users are not satisfied in general.

Fortunately, various kinds of auxiliary information have become increasingly available in online services. This auxiliary information can be easily organized into heterogeneous information networks. Heterogeneous information networks contain rich attribute information and semantic associations, so can provide potential relations between users and items for recommender systems [7]. By connecting different kinds of relations in heterogeneous information networks, latent higher-order interaction information between users and items can be discovered. The emerging success of mining heterogeneous information networks may shed some light on solving these issues of data sparseness and simple explanation in the recommendation system. Many existing models [8] regard reviews, an item's aspects and meta-paths as contextual information about the user–item interaction and leverage them to improve the recommendation performances and generate recommendation explanations. Explainable recommendation has also attracted remarkable attention in recent years [9,10].

Although the above methods have achieved a better performance, there are two challenges in applying heterogeneous information networks to recommender systems: (1) how to extract effective information that can be used in recommendation systems from heterogeneous information networks; (2) how to effectively integrate high-order interaction information for better recommendation results and explanations. In order to solve the first challenge, we intend to design multiple different types of meta-paths for heterogeneous information network architectures to produce corresponding similarity matrices [11]. As for auxiliary information, it can tackle the issue of the sparseness of the original user–item interaction matrix. Then, the latent representations of users and items are obtained through matrix decomposition methods [12]. Aiming at the second issue, we present a dual-attention network to distinguish the contribution of each representation from different meta-paths to the final representations of users and items. Then, the dual-attention networks will aggregate the representations from multiple meta-paths through the attention coefficients to generate the final representations of users and items.

In this paper, we propose a framework of explainable recommendation by exploiting dual-attention networks in heterogeneous information networks (DANER), to capture the latent representations of user preferences and item features, and to learn the joint representation of user–item interactions using the dual-attention networks for the recommendation predictions and explanations. The contributions of this paper are summarized as follows:

- In order to alleviate the problem of data sparseness, we extracted multiple kinds of meta-paths between users and items from the heterogeneous information networks and generated multiple similarity matrices, which were used as complements of the rating matrix. Then, we decomposed the similarity matrices by matrix decomposition to obtain the multiple representations of users and items corresponding to different meta-paths;
- We propose a novel dual-attention network for explainable recommendation in heterogeneous information networks (DANER). It leverages a local attention layer to learn the representations of users and items, and a global attention layer to learn the joint representations of user–item interactions, both of which integrate multiple groups of different meta-path information. An attention mechanism helps to improve the explainability of the recommendation;
- We demonstrate better rating prediction accuracy than the state-of-the-art methods by performing comprehensive experiments on two benchmark datasets. In addition,

by providing a critical meta-path based on attention coefficient, we show a case study on the explainability of DANER.

The rest of this paper is organized as follows: Section 2 highlights the related work of typical recommendation methods; HIN in recommendation and attention mechanisms, respectively. Section 3 introduces the definition and problem formulation. Section 4 presents the details of our proposed DANER model. Section 5 shows the experimental results. Finally, Section 6 concludes this paper.

2. Related Works

2.1. Recommendation

At present, there are two main streams of recommendation system models, one is based on collaborative filtering and the other is based on content [13]. The core idea of collaborative filtering model is to recommend items for users according to the preferences of a group of users with similar interests and common experiences [14]. Collaborative filtering recommendation algorithms can be divided into two categories: user-based collaborative filtering and item-based collaborative filtering. Matrix factorization (MF) [15] is one of the most widely used methods, which factorizes a high-dimensional original matrix into two low-dimensional matrices and uses the two new matrices to calculate the prediction rating. There are many extended works based on matrix factorization, including Probability Matrix Factorization [16], BPRMF [17]. The Probabilistic Matrix Factorization (PMF) model scales linearly with the number of observations and performs well on the large, sparse, and very imbalanced Netflix datasets, including an adaptive prior on model parameters, which shows how the model's capacity can be controlled automatically. BPRMF presents a generic optimization criterion BPR-Opt for personalized ranking that is the maximum posterior estimator derived from a Bayesian analysis of the problem, which also provides a generic learning algorithm for optimizing models with respect to BPR-Opt. Factorization Machines (FMs) [18] introduce a new model class that combines the advantages of Support Vector Machines with factorization models. It is a general predictor working with any real valued feature vector, which model all interactions between variables using factorized parameters. SVD++ [19] introduces a new neighborhood model with an improved prediction accuracy, which models neighborhood relations by minimizing a global cost function.The new neighborhood model adds a global average rating, item rating bias, user rating bias, and interest preference between user and item. Content-based recommendation mainly generates recommendations based on item characteristics and user profiles. The factorization machine is the typical representative among them, mainly to solve the problem of feature combination under the condition of sparse data.

2.2. HIN in Recommendation

Heterogeneous information networks have been proposed as a general representation of a graph or network in many real world scenarios [20,21]. Because of their remarkable ability to represent heterogeneous data, they have been used in a large number of tasks of data mining, such as clustering, classification, link prediction, representation learning and similarity measurement [22,23]. Recently, some recommendation systems have used heterogeneous information networks as auxiliary information, which has achieved great success [24]. In heterogeneous information networks, a sequence composed of different types of nodes is defined as a meta-path, which is capable of extracting the relation information [25,26]. HeteMF [27] is a matrix factorization based model which takes advantage of both rating data and the related information network, and uses meta-path-based similarity as a regularization term to enhance the effect of the recommendation model. HeteRec [28] proposes to combine heterogeneous relationship information for each user differently and introduces meta-path-based latent features to represent the connectivity between users and items along different types of paths. SemRec [29] is the first to propose weighted HIN and weighted meta-path concepts to

subtly depict the path semantics through distinguishing different link attribute values. SemRec not only flexibly integrates heterogeneous information but also obtains prioritized and personalized weights representing user preferences on paths. FMG [30] first introduces the concept of the meta-graph to HIN-based recommendation and then solves the information fusion problem with a "matrix factorization (MF) + factorization machine (FM)" approach.

2.3. Attention Mechanism

When human beings observe a scene, they always focus on some objects in the scene according to the guidance of the information they want to obtain. Inspired by this, researchers introduced an attention mechanism into machine learning and achieved remarkable results [31]. The core purpose of the attention mechanism is to select the most critical information for the current task from a large amount of information [32,33]. Nowadays, the attention mechanism has been widely used in various fields of deep learning, such as image processing, speech recognition, natural language processing, and recommendation [34–36]. ACF [37] introduced a novel attention mechanism in CF to address the challenging item-level and component-level implicit feedback in the recommendation. The model consists of two attention modules—the component-level attention module to select informative components of multimedia items and the item-level attention module to score item preferences. Graph attention networks (GATs) [38] present a novel neural network architecture that operates on graph-structured data, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations. KGAT [39] proposes a new method named Knowledge Graph Attention Network that explicitly models the high-order connectivities in KG in an end-to-end fashion. It recursively propagates the embeddings from the node's neighbours to refine the node's embedding, and employs an attention mechanism to discriminate the importance of the neighbours. HGAT [40] first proposed a novel heterogeneous graph neural network based on hierarchical attention, including node-level and semantic-level attentions, in which the contribution of node and meta-path can be fully considered.

3. Problem Statement

3.1. Definitions

There are several definitions of heterogeneous information networks. In this paper, we introduce the definitions of HIN, the network schema and the meta-path. Next, we will illustrate these three definitions in detail.

Definition 1 (Heterogeneous Information Networks)**.** *HIN is defined as a graph G=(V,E) with an object type mapping function $\varphi : V \rightarrow A$ and a relation type mapping function $\psi : E \rightarrow R$, where each object $v \in V$ belongs to a specific object type $\varphi(v) \in A$, and each relation $e \in E$ corresponds to a specific relation type $\psi(e) \in R$, where the number of object types $A > 1$ or relation types $R > 1$.*

An example of the heterogeneous information networks is shown in Figure 1. There are four object types and three relation types in the heterogeneous information networks. The four object types are group, user, business and category. The relation between group and user indicates that a user belongs to a group. The relation between user and business indicates that a user prefers a business. The relation between business and category indicates that a business belongs to a category.

Definition 2 (Network Schema)**.** *The network schema is a meta template of heterogeneous networks G=(V,E) including object mapping function $\varphi : V \rightarrow A$ and relation type mapping function $\psi : E \rightarrow R$. Network schema is defined as a directed graph composed of object types A and relation types R, denoted as $T_G = (A, R)$.*

Figure 2 illustrates the network schema corresponding to the Yelp dataset. The Yelp dataset has five object types and five relation types. There may be more than one meta-path between two objects in an HIN. For example, user and business can be connected via $U \rightarrow B \leftarrow U \rightarrow B$ or $U \rightarrow B \rightarrow Cate \leftarrow B$. These paths are called meta-paths, defined in Definition 3.

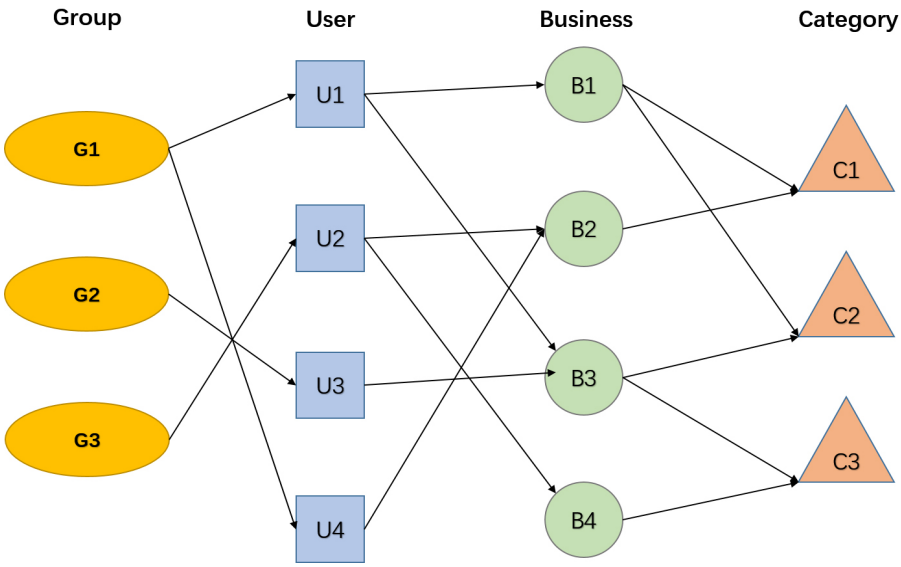

Figure 1. A toy example of HIN.

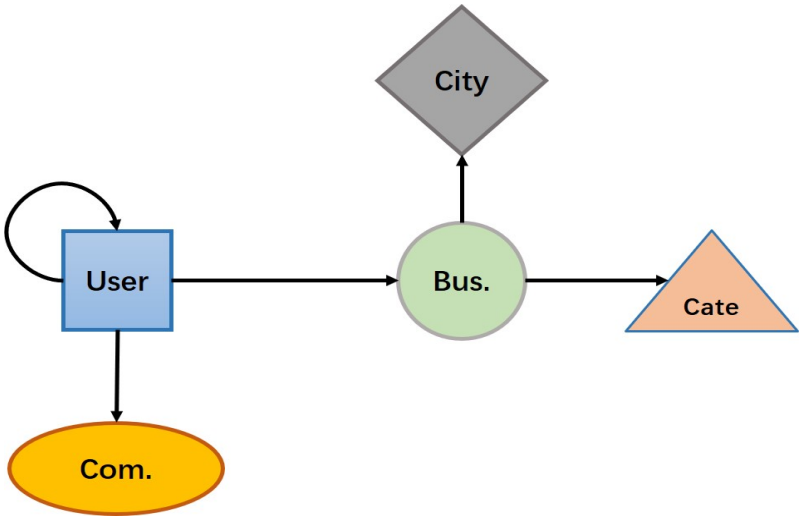

Figure 2. Network schema of Yelp dataset.

Definition 3 (Meta-path). *A meta-path is a path defined in the network schema with a starting node and a target node, such as* $A_1 \xrightarrow{r_1} A_2 \xrightarrow{r_2} \ldots A_{n-1} \xrightarrow{r_{n-1}} A_n$, *where* $A_i \in A$ *is the type of different object and* $r_i \in R$ *is the relation between the two objects. Apparently, the complex relation between node* A_1 *and node* A_n *can be represented in meta-path, denoted as* $R = r_1 \circ r_2 \ldots \circ r_n$, *where the number of relation* R_i *is the length of the meta-path.*

3.2. Problem Statement

For inputs to our framework, we have the user set $U = \{u_1, u_2, \ldots, u_U\}$, the business set $B = \{b_1, b_2, \ldots, b_B\}$, and the relation set $R = \{r_1, r_2, \ldots, r_R\}$, where r_i is the relation between two objects which can be user, business, category, city and so on. When the r_i represents the relation of user and business, the weight between them indicates the rating of user on business. We design multiple meta-paths MP_i, and obtain multiple similarity

matrices M_i through the meta-paths. For the output of our framework, we provide the predicted rating $\hat{r}_{ui}$ of user on business and a meta-path-level explanation.

Accordingly, the two main tasks of DANER can be summarized as: (1) obtaining more expressive presentations of user preferences and item features through auxiliary information in heterogeneous information networks; (2) using the attention mechanism to aggregate these representations to get better recommendation results and providing some explanations based on the attention coefficient simultaneously.

4. Framework

In this section, we mainly introduce the use of auxiliary information in HIN and the establishment of recommendation model based on double attention mechanism. The overall structure of DANER is shown in Figure 3. DANER mainly includes three parts: the Similarity Matrices Generation, the Matrix Decomposition and the Recommendation model based on Attention Mechanism.

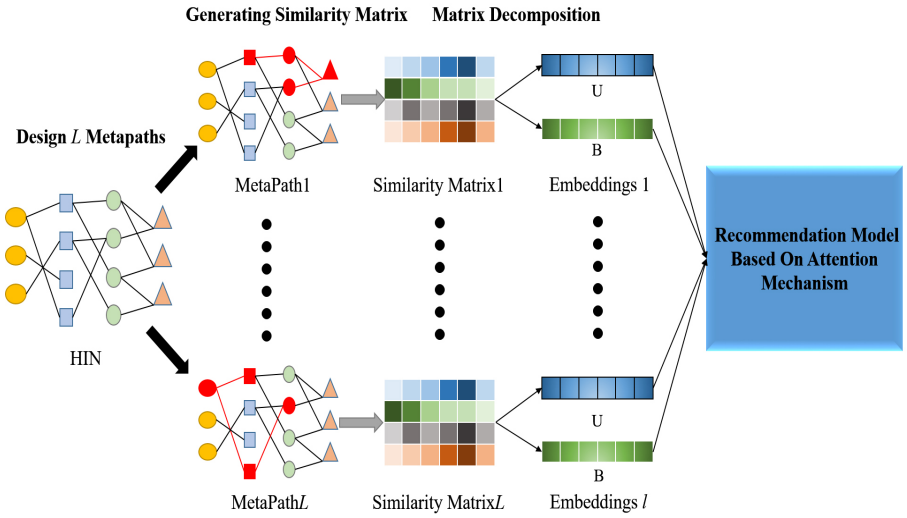

Figure 3. The overall structure of DANER.

4.1. Similarity Matrix Based on Meta-Path

For a recommendation system, the starting node of the meta-path is user u, and the target node is item i. The meta-path MP_{ui} represents the high-order relation between user u and item i. For example, the meta-path $u_1 \xrightarrow{Buy} i_1 \xrightarrow{BelongTo} Cate_1 \xleftarrow{BelongTo} i_2$ in Amazon dataset indicates that user u_1 has purchased item i_1, and item i_1 and item i_2 belonging to the same category. The similarity matrix based on the meta-path is defined as $M_{MP} = M_{A_1A_2} \otimes M_{A_2A_3} \ldots \otimes M_{A_{n-1}A_n}$, where $M_{A_{n-1}A_n}$ represents the relation matrix between the object type A_{n-1} and A_n, $\otimes$ is a matrix multiplication operation between the two relation matrices.

L user–item similarity matrices can be obtained by L pre-designed meta-paths. The meta-paths of different datasets used in the experiment are shown in Table 1. For example, the meta-path used in the dataset Amazon $U \rightarrow B \leftarrow U \rightarrow B$ indicates that users will buy other items that have been purchased by users with the same preferences, which can be regarded as a user-based collaborative filtering. Thus, the similarity matrix corresponding to this meta-path can be obtained by the following formula $M_{UBUB} = M_{UB} \otimes M_{BU} \otimes M_{UB}$. Besides, *Cate* refers to the categories to which the item belongs, *Brand* refers to the brand to which the item belongs, *View* refers to which other items have been viewed by users who have rated the item, *Com* refers to the number of compliments a user receives from other users, and *City* refers to the city in which the restaurant is located. The calculation of the relevant meta-path is similar to the procedure mentioned above.

Table 1. Meta-paths defined in different datasets.

Dataset	Meta-Path
Amazon	$U \rightarrow B$ $U \rightarrow B \leftarrow U \rightarrow B$ $U \rightarrow B \rightarrow Cate \leftarrow B$ $U \rightarrow B \rightarrow Brand \leftarrow B$ $U \rightarrow B \rightarrow View \leftarrow B$ $U \rightarrow B \rightarrow Cate \leftarrow B \leftarrow U \rightarrow B$ $U \rightarrow B \rightarrow Brand \leftarrow B \leftarrow U \rightarrow B$ $U \rightarrow B \rightarrow View \leftarrow B \leftarrow U \rightarrow B$
Yelp	$U \rightarrow B$ $U \rightarrow U \rightarrow B$ $U \rightarrow B \leftarrow U \rightarrow B$ $U \rightarrow Com \leftarrow U \rightarrow B$ $U \rightarrow B \rightarrow Cate \leftarrow B$ $U \rightarrow B \rightarrow City \leftarrow B$ $U \rightarrow B \rightarrow Cate \leftarrow B \leftarrow U \rightarrow B$ $U \rightarrow B \rightarrow City \leftarrow B \leftarrow U \rightarrow B$

4.2. Latent Representation by Matrix Decomposition

The recommendation learning process can be regarded as a representation learning process [41]. After obtaining user–item similarity matrices corresponding to L meta-paths, we adopted matrix decomposition to obtain the latent representations of users and items. By using low-dimensional vector of the latent representation, we can reduce noise and alleviate the data sparseness problem of the original rating matrix [42]. Based on the theory of matrix decomposition, the similarity matrix M can be decomposed into two low rank matrices I_U and I_B, where I_U represents the latent features of users' preferences and I_B represents the latent features of items. Then we can use $\hat{M} = I_U \times I_B$ to generate the prediction similarity matrix $\hat{M}$. By reducing the difference between M and $\hat{M}$, we can obtain the latent representation matrices I_U and I_B, which can represent the latent features of users and items better. To be specific, low-dimensional representations of users and items can be obtained by solving the following optimization problem:

$$\min_{U,B} (\hat{M} - M)^2 + \lambda_1 \|I_U\|_F^2 + \lambda_2 \|I_B\|_F^2, \tag{1}$$

where λ_1 and λ_2 are dynamic parameters, which are used to control the influence of Frobenius norm regularization to avoid overfitting. The goal of optimization is to make I_U and I_B restore the similarity matrix M as complete as possible.

For L similarity matrices based on meta-paths, we can obtain L groups of feature representations of users and items $I_U^{(1)}, I_B^{(1)}, I_U^{(2)}, I_B^{(2)} \ldots I_U^{(L)}, I_B^{(L)}$ by performing a matrix decomposition operation.

4.3. Recommendation Model Based on Attention Mechanism

After obtaining L groups of representations of users and items, we also need to fuse them to obtain more expressive representations of users and items. Thus, at first, we designed a model including two attention networks to integrate these representations. The local attention network was oriented to each user (item), which was used to distinguish the importance of each user (item) representation corresponding to different meta-paths. According to the weighted combination of attention coefficients, the representations of users and items integrating L groups of meta-path information can be obtained respectively. The global attention network is oriented to each meta-path, which is capable of discriminating the importance of each user–item joint representation corresponding to different meta-paths. Besides, the attention coefficients can be used to select the meta-path that has the most influence on the final prediction results. By way

of the global attention network, we can obtain the user–item joint representations integrating L groups of meta-path information. Then, the representations obtained from the two attention networks are concatenated as the input of next part. Finally, we can utilize a multi-layer perceptron to generate the prediction ratings. The specific recommendation model is shown in Figure 4, mainly including three parts, which will be introduced separately below.

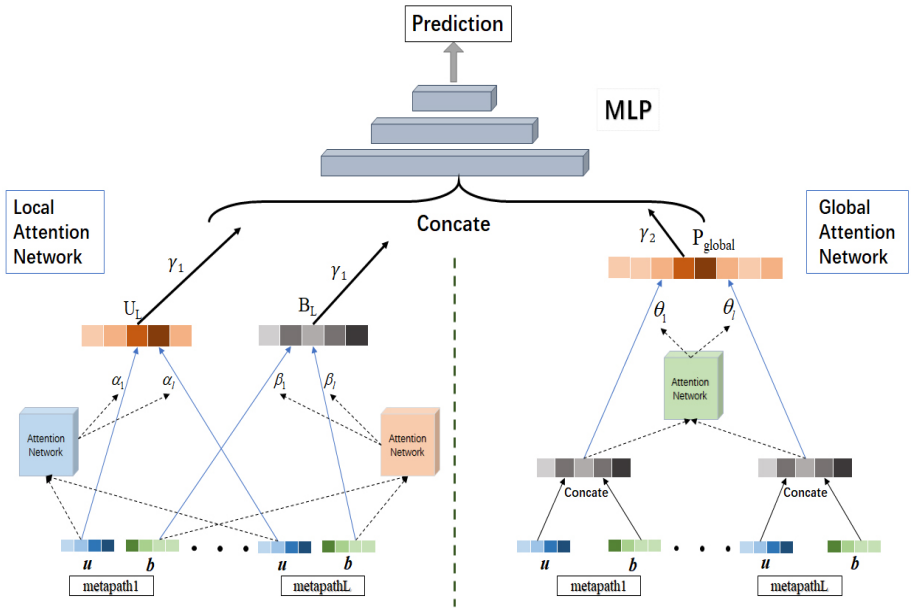

Figure 4. Recommendation model based on attention mechanism.

4.3.1. Local Attention Network

The goal of local attention network is to learn the representations of users and items, which integrate L groups of representations corresponding to different meta-paths. The input of local attention network is L groups of representations of user and item obtained by matrix decomposition. Each group of representations contains user representation $I_U^{(i)}$ and item representation $I_B^{(i)}$. For L groups of user representations $I_U^{(i)}$ $(i = 1, \ldots, L)$, we feed them into the user-oriented attention neural network to obtain the attention coefficient α_i corresponding to $I_U^{(i)}$:

$$DNNu_i = Relu(\mathbf{W}_U^{(n)} \times \cdots \times Relu(\mathbf{W}_U^{(1)} \times I_U^{(i)} + \mathbf{B}_U^{(1)}) + \cdots + \mathbf{B}_U^{(n)}), \tag{2}$$

$$\alpha_i = \frac{e^{DNNu_i}}{\sum_{i \in L} e^{DNNu_i}}, (i = 1, \ldots, L), \tag{3}$$

where $DNNu$ is a user-oriented attention neural network. To be specific, the input of $DNNu$ is user representations from different meta-paths, and its output is attention scores. $\mathbf{W}_U^{(i)}$ and $\mathbf{B}_U^{(i)}$ are parameter matrix and bias term of the fully connected neural network of layer i, and we use $Relu$ as the activation function of each layer. Then, to compute the attention coefficient α_i, the $Softmax$ function is introduced to normalize L output values of the neural network.

By adopting the same operation for item, the attention coefficient β_i corresponding to the item $I_B^{(i)}$ $(i = 1, \ldots, L)$ from different meta-paths can be obtained as follows:

$$DNNb_i = Relu(\mathbf{W}_B^{(n)} \times \cdots \times Relu(\mathbf{W}_B^{(1)} \times I_B^{(i)} + \mathbf{B}_B^{(1)}) + \cdots + \mathbf{B}_B^{(n)}) \tag{4}$$

$$\beta_i = \frac{e^{DNNb_i}}{\sum_{i \in L} e^{DNNb_i}}, (i = 1, \ldots, L). \tag{5}$$

Then, according to the obtained attention coefficients α_i and β_i, we can combine L groups of representations of user (item) from different meta-paths to produce U_L (B_L). The adopted combination method is to multiply the user (item) representation with the corresponding attention coefficient $\alpha_i(\beta_i)$, and then directly concatenate the L groups of $\alpha_i \times I_U^{(i)}(\beta_i \times I_B^{(i)})$:

$$U_L = Concate(\alpha_1 \times I_U^{(1)}, \alpha_2 \times I_U^{(2)} \ldots, \alpha_L \times I_U^{(L)}) \tag{6}$$

$$B_L = Concate(\beta_1 \times I_B^{(1)}, \beta_2 \times I_B^{(2)} \ldots, \beta_L \times I_B^{(L)}). \tag{7}$$

The local attention network layer can generate user representation U_L and item representation B_L, which contain different meta-path information and focus on the critical meta-path information. The degree of reservation of meta-path information depends on the value of attention coefficient, the larger the attention coefficient is, the more meta-path information will be retained.

Finally, we can concatenate the user representation U_L and item representation B_L to obtain the local user–item joint representation P_{local}, which is a part of the input vector of multi-layer perceptron in the interaction model, as shown below:

$$\begin{aligned} P_{local} &= Concate(U_L, B_L) \\ &= (\alpha_1 \times I_U^{(1)} \alpha_2 \times I_U^{(2)} \ldots \alpha_L \times I_U^{(L)} \\ &\qquad \beta_1 \times I_B^{(1)} \beta_2 \times I_B^{(2)} \ldots \beta_L \times I_B^{(L)}). \end{aligned} \tag{8}$$

4.3.2. Global Attention Network

The global attention network focuses on distinguishing the contributions of user–item joint representations corresponding to different meta-paths. Firstly, we concatenate the representations $I_U^{(i)}$ and $I_B^{(i)}$ to obtain the user–item joint representation $p_{joint}^{(i)}$, where $p_{joint}^{(i)} = I_U^{(i)} I_B^{(i)}$ $(i = 1, \ldots, L)$. Then, we feed the L groups of $p_{joint}^{(i)}$ into the path-oriented neural network $DNNp$ to compute the corresponding attention coefficient θ_i:

$$DNNp_i = Relu(\mathbf{W}_P^{(n)} \times \cdots \times Relu(\mathbf{W}_P^{(1)} \times p_{joint}^{(i)} + \mathbf{B}_P^{(1)}) + \cdots + \mathbf{B}_P^{(n)}) \tag{9}$$

$$\theta_i = \frac{e^{DNNp_i}}{\sum_{i \in L} e^{DNNp_i}}, (i = 1, \ldots, L), \tag{10}$$

where $\mathbf{W}_P^{(i)}$ and $\mathbf{B}_P^{(i)}$ are parameter matrix and bias terms of the fully connected neural network of layer i, the input of $DNNp$ is L groups of user–item joint representations p_{joint}, the output is attention scores. Besides, $Relu$ is used as the layer activation function in the neural network. After that, to obtain the attention coefficients θ, we introduce the $Softmax$ function to normalize the L output values of the neural network.

Finally, according to the obtained attention coefficients θ, we combine the user–item joint representations p_{joint} from L groups of meta-paths to obtain the global user–item joint representation P_{global}. Here, we propose to multiply the L groups of user–item joint representations p_{joint} with the corresponding attention coefficients θ. Then, we concatenate L groups of $\theta_i \times p_{joint}^{(i)}$ to obtain P_{global} directly:

$$P_{global} = Concate(\theta_1 \times p_{joint}^{(1)}, \theta_2 \times p_{joint}^{(2)} \ldots, \theta_L \times p_{joint}^{(L)}). \tag{11}$$

Based on the attention coefficients of global attention network, we can explain the recommendation results more sufficiently, that is, the meta-path with large attention coefficient contributes more to the recommendation result.

4.3.3. Interaction Model

After obtaining the local user–item joint representation P_{local} by way of the local attention network and the global user–item joint representation P_{global} by global attention network, respectively, we need to integrate them together as the input of the subsequent interaction model [43,44]. Here are two kinds of combination methods:

$$P^1_{total} = \gamma_1 \times P_{local} + \gamma_2 \times P_{global} \tag{12}$$

$$P^2_{total} = Concate(\gamma_1 \times P_{local}, \gamma_2 \times P_{global}), \tag{13}$$

where $\gamma_1 \in (0, 1)$ and $\gamma_2 \in (0, 1)$ are weighted parameters of P_{local} and P_{global}. The first method is to add the local user–item joint representation P_{local} and the global user–item joint representations P_{global} weighted by γ_1 and γ_2, and the second method is to use concatenation instead of addition in the first method. Based on these two methods, we design two variants of the model in the experiment section. Both add and concat are common operations used to aggregate feature information in neural networks. The concate operation overlays the dimensions of the feature vector. The information contained in each dimension of the vector does not change, but the dimension of the vector is doubled. The add operation adds the corresponding values of the feature vectors. The dimensions of the vectors do not change, but the information contained in each dimension is increased. Add enriches the representation information for each feature, while concat increases the number of features. After obtaining the combined user–item joint representation P_{total}, we need an interaction model to fuse the feature information of representation for generating the rating prediction. The traditional methods mostly use Factorization Machine, which has the advantages of simple operation and low calculation cost. But it can only fuse the first-order and second-order features. So it is difficult to fuse the high-order features. Therefore, in this paper, multi-layer perceptron is adopted as the interaction model, due to its powerful capability of automatically combining high-order features. What is more, the input of multi-layer perceptron model is combined with user–item joint representation P_{total}, the output is the prediction rating, defined as follows:

$$y_{pred} = Relu(\mathbf{W}^{(n)} \times \cdots (Relu(\mathbf{W}^{(1)} \times P_{total} + \mathbf{B}^{(1)}) + \cdots \mathbf{B}^{(n)}). \tag{14}$$

4.4. Model Optimization

The task of this paper is rating prediction based on explicit data. Here, the square loss function is used as the optimization goal [17]:

$$Loss = (y_{pred} - y_{real})^2 + \epsilon \times \|Para\|^2, \tag{15}$$

where y_{pred} is the prediction rating obtained by the proposed framework, y_{real} is the real rating of user on item, $Para$ are the trainable parameters in the neural network. The first term indicates the difference between y_{pred} and y_{real}, and the second term is L_2 norm regularization, in which the coefficient ϵ is devised to control the regularization intensity to prevent overfitting.

5. Experiments

5.1. Datasets

To verify the effectiveness of DANER, we performed experiments on two real datasets with rich heterogeneous information. The first dataset was Amazon (https://nijianmo.github.io/amazon/index.html#files accessed on 14 November 2022), which contains reviews (ratings, text, votes), item data (item description, item type, price, brand and image characteristics) and purchase relations from the Amazon e-commerce site. The second dataset is Yelp (https://www.yelp.com/dataset accessed on 14 November 2022), which con-

tains reviews, city information, item attributes, and user characteristics from the American review site Yelp.

The detailed statistics of the datasets used in this paper are shown in Table 2. The Amazon dataset contains 195,791 rating data for 6170 users and 2753 items, and the Yelp dataset contains 19,397 rating data for 16,239 users and 14,284 items. The value of rating ranges from 1 to 5. The higher the value of the rating is, the more attention the user pays to the item. While the lower the rating is, the less interest the user has in the item. All relations used to construct meta-paths have been extracted from the formatted strings in the two original datasets. We use data density to measure data sparseness—the data density calculated as follows:

$$Density = \frac{num(ratings)}{num(users) \times num(items)}. \tag{16}$$

Table 2. Statistics of Amazon and Yelp datasets.

Amazon		**Yelp**	
Entity	**Number**	**Entity**	**Number**
User	6170	User	16,239
Item	2753	Business	14,284
View	3857	Compliment	11
Category	22	Category	511
Brand	334	City	47
Relation	Number	Relation	Number
user–item	195,791	User-Business	198,397
Item-View	5694	User-User	158,590
Item-Categoty	5508	User-Compliment	76,875
Item-Brand	2753	Business-City	14,267
		Business-Category	40,009
Density = 1.15%		Density = 0.086%	

5.2. Evaluation Metrics

In order to accurately evaluate the performances of DANER and baseline methods, we adopted two widely used metrics—Mean Absolute Error (MAE) and Rooted Mean Square Error $(RMSE)$—as the evaluation metrics of our experiment. MAE is the average of absolute error between the prediction value and the real value, which can accurately reflect the actual prediction error. $RMSE$ is the square root of the ratio of the deviation between the prediction value and the real value over the number of samples. $RMSE$ indicates the dispersion of data, which is commonly used as a standard metric for prediction tassk in the machine learning model. MAE and $RMSE$ are defined as follows:

$$MAE = \frac{1}{|R_{test}|} \sum_{(i,j) \in R_{test}} |\hat{R_{i,j}} - R_{i,j}|. \tag{17}$$

$$RMSE = \sqrt{\frac{\sum_{(i,j) \in R_{test}} (\hat{R_{i,j}} - R_{i,j})^2}{|R_{test}|}}, \tag{18}$$

where R_{test} is a test set of user–item interaction records, R_{test} is the number of user–item interaction records in R_{test}, $\hat{R_{i,j}}$ is the prediction rating obtained by DANER, and $R_{i,j}$ is the real rating of user on item. The smaller the values of MAE and $RMSE$ are, the better the recommendation performance is.

5.3. Baselines

In order to verify the performance of the proposed framework, the following baseline methods were chosen for comparison.

RegSVD [15]: RegSVD belongs to collaborative filtering method based on users, of which the input is a single user–item rating matrix. It adds L_2 norm regularization term to constrain the representation vectors of users and items based on matrix decomposition method.

SVD++[19]: The SVD++ model adds a global average rating, user rating deviation, item rating deviation and user history rating information into the optimization objective function, which achieves a significant improvement on overcoming the problem that the original SVD method does not explicitly consider the impact of a user's historical behavior on user rating prediction. At present, it has become one of the most typical methods in the field of recommender systems.

NeuCF [45]: NeuCF creatively introduces deep learning into recommendation system and overcomes the shortcomings of inner product operation by using deep neural network as the interaction model between users and items. In addition, NeuCF combines linear and non-linear interaction models to recommend items based on implicit data. In order to apply NeuCF to the rating prediction task of explicit data, the pairwise loss function of the model is replaced by the square error loss function on the basis of the original paper code. Also, the activation function of the neural network is changed to the *Relu* function.

FMG [30]: FMG introduces heterogeneous information networks as auxiliary information for a recommendation task. In particular, it uses meta-graph to mine the information of heterogeneous networks and adopts matrix decomposition to obtain representations of users and items. Besides, Factorization Machine with group lasso regularization is employed to generate recommendation results. Because the difference of the values in the generated similarity matrices is too large, we add a standardized operation before the matrix decomposition to keep the similarity values in the range of 1 to 5.

5.4. Experimental Performance

5.4.1. Experimental Settings

In the experiments, we used Python and TensorFlow deep learning framework to implement the proposed DANER model. In the comparison experiments of model variations and baseline methods, to obtain the optimal experimental performance, the dimension of representation vectors is fixed to 32 for all models. The learning rates and regularization coefficients are tuned in 0.001, 0.005, 0.01, 0.05, 0.1 according to different models. Besides, the drop out ratio is set to 0.5, and the batch size is set to 256.

The parameters of the model were initialized by the uniform distribution initializer. We optimized the model with the Adaptive Moment Estimation optimizer [46]. We also designed an early stopping mechanism to control the training time. When the evaluation metrics or train data loss does not decrease for 20 successive epochs, the training process can be terminated. In order to compare the performances of the models with different training sets, we utilize (80%, 70%, 60%) of Amazon and Yelp datasets for training and the remaining (20%, 30%, 40%) for testing.

5.4.2. Model Variations Comparison Experiments

There are many optional model variations in the modelling process. Therefore, a variety of model variations comparison experiments were designed to find out the best model variation as the final model for the subsequent comparison experiments of baseline methods. Furthermore, through the model variations comparison experiments, we can verify whether the attention mechanism is helpful for improving the performance of recommendation, and can even figure out how the attention mechanism finds the critical meta-path.

In the experiments, five variations of the model with different combination operations on fusing the local and the global user–item joint representations are designed, including the no-attention model (DANER-no), the local-attention model (DANER-local), the global-

attention model (DANER-global), the adding-attention model (DANER-add) and the concatenating-attention model (DANER-concate), respectively. The DANER-no does not contain any attention mechanism, and the representations of users and items obtained by matrix decomposition are fed into the interaction model directly. The DANER-local only uses the local attention network after matrix decomposition to obtain the local user–item joint representations as the input of the interaction model. Similar to DANER-local, the DANER-global only uses global attention network between matrix decomposition and interaction model. The DANER-add adopts adding operation when combining the user–item joint representations P_{local} and P_{global}. The DANER-concate adopts concatenating operation instead of adding operation in DANER-add. The results of the model variations comparison experiments are shown in Tables 3 and 4.

Table 3. Experimental results MAE of model variations comparison experiments. The best results are highlighted in boldface.

Dataset/Ratio	DANER-No	DANER-Local	DANER-Global	DANER-Add	DANER-Concate
Amazon/20%	0.819	0.691	0.69	0.687	**0.672**
Amazon/30%	0.823	0.697	0.696	0.693	**0.689**
Amazon/40%	0.849	0.70	0.701	0.698	**0.695**
Yelp/20%	0.837	0.793	0.792	0.799	**0.784**
Yelp/30%	0.858	0.795	0.805	0.85	**0.795**
Yelp/40%	0.86	0.799	0.809	0.809	**0.798**

Table 4. Experimental results RMSE of model variations comparison experiments. The best results are highlighted in boldface.

Dataset/Ratio	DANER-No	DANER-Local	DANER-Global	DANER-Add	DANER-Concate
Amazon/20%	1.066	0.941	0.937	0.935	**0.934**
Amazon/30%	1.071	0.947	0.945	0.945	**0.944**
Amazon/40%	1.075	0.949	0.948	0.95	**0.946**
Yelp/20%	1.065	1.017	1.015	1.016	**1.009**
Yelp/30%	1.073	1.024	1.03	1.025	**1.019**
Yelp/40%	1.08	1.029	1.031	1.028	**1.022**

According to the experimental results, it is apparent that the performance of DANER-no is the worst over all the variations, which shows that the attention mechanism is helpful to improve the recommendation results. There is no significant gap between the four kinds of models with attention mechanism. However, the performances of DANER-add and DANER-concate are better than those of DANER-local and DANER-global. This may be due to the fact that the latter can mine more feature information of users and items. We can also observe that the DANER-concate outperforms the DANER-add, which shows that the DANER-concate is the best among the five variations. The reason may be that concatenating operation can preserve the feature information of users and items more effectively. Therefore, we use the DANER-concate as the final method of our proposed framework and compare it with the baseline methods. After taking the whole experimental results into account, we can come to the conclusion that the introduction of attention mechanism does improve the accuracy of rating prediction, both of the $RMSE$ and MAE metrics decreasing by more than 5%.

5.4.3. Baseline Methods Comparison Experiments

In this section, we will compare our DANER-concate model with the baseline methods. The results are shown in Tables 5 and 6.

Table 5. Experimental results MAE of baseline methods comparison experiments. The best results are highlighted in boldface.

Dataset/Ratio	RegSVD	SVD++	NeuCF	FMG	DANER
Amazon/20%	0.728	0.715	0.702	0.711	**0.672**
Amazon/30%	0.731	0.726	0.722	0.717	**0.689**
Amazon/40%	0.749	0.781	0.739	0.722	**0.695**
Yelp/20%	0.833	0.818	0.808	0.796	**0.784**
Yelp/30%	0.835	0.819	0.814	0.81	**0.795**
Yelp/40%	0.841	0.825	0.828	0.819	**0.798**

Table 6. Experimental results RMSE of baseline methods comparison experiments. The best results are highlighted in boldface.

Dataset/Ratio	RegSVD	SVD++	NeuCF	FMG	DANER
Amazon/20%	0.957	0.949	0.954	0.947	**0.934**
Amazon/30%	0.961	0.954	0.957	0.949	**0.944**
Amazon/40%	0.986	0.966	0.963	0.954	**0.946**
Yelp/20%	1.066	1.051	1.027	1.025	**1.008**
Yelp/30%	1.068	1.054	1.032	1.032	**1.019**
Yelp/40%	1.075	1.060	1.037	1.034	**1.022**

By observing the experimental results on Amazon and Yelp datasets, it can be found that the larger the proportion of training set is, the smaller the MAE and $RMSE$ metrics of the experiments are, which indicates that the prediction rating is more accurate. This is because the performance of the recommendation task is greatly affected by the sparseness of the rating matrix. As shown in Table 2, the density of the original datasets are very small, and the rating matrices are very sparse. When the proportion of training data increases, the rating matrix of the training set becomes denser. So more rating information can be obtained, which leads to a better performance.

As we can see in Tables 5 and 6, RegSVD and SVD++ have the worst performance under six experimental conditions of two datasets. They are traditional machine learning methods, which neither use neural networks nor contain auxiliary information of heterogeneous information networks. In addition, SVD++ performs better compared to RegSVD, which may be credited with the fact that SVD++ contains more user history information. In most of the experimental results, NeuCF obviously outperforms the former two methods, except the $RMSE$ metric on the Amazon dataset. It is a deep learning model, which uses neural network as the interaction model to overcome the shortcomings of the previous inner product operation. But because the model is designed for the top-N recommendation task of implicit data, it may not be able to play its effect on the rating prediction task of explicit data completely. FMG yields better performance than the previous three methods (RegSVD, SVD++ and NeuCF) which only use the rating information of users and items. Moreover, FMG utilizes meta-graph to extract the information of heterogeneous information networks as auxiliary information, which can help to solve the problem of sparse original rating matrix to a certain extent and get better recommendation performance. In general, the proposed framework DANER surpasses the baseline methods consistently over all conditions in the experiments. The improvement of MAE metric is more remarkable than that of $RMSE$ metric. Specifically, MAE metric of DANER increases by 3.7–4.3% on Amazon dataset and 1.5–2.6% on Yelp dataset. $RMSE$ metric of DANER increases by 0.4–0.8% on Amazon dataset and 1.2–1.7% on Yelp dataset. The reasons for the progress of our framework are as follows: (i). we introduce heterogeneous information networks as auxiliary information to alleviate the data sparseness problem of single rating matrix; (ii). we utilized an attention mechanism to make the generated representations more abundant and effective.

5.4.4. Parameter Comparison Experiments

In this section, we designed a series of comparison experiments to study the influence of K on the experimental results, where K is the dimension of representation vector in matrix decomposition. The value of the dimension K affects the amount of information contained in the representation vector. To be specific, the larger the K value is, the better the expressive power of the representation vector is. However, larger K value will lead to more space consumption and increase the calculation cost of matrix decomposition at the same time. Therefore, in order to get the appropriate K value which can balance the performance of the model and the computational cost, we conducted parameter comparison experiments on Amazon and Yelp datasets, where the K value waas set to 8, 16, 32 and 64 respectively. The results of the parameter comparison experiments are shown in Figures 5 and 6.

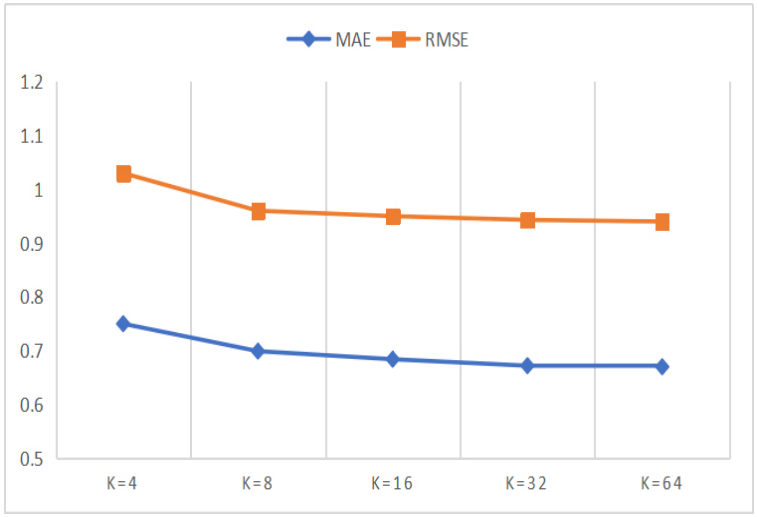

Figure 5. The performances of the DANER model with different K on Amazon dataset.

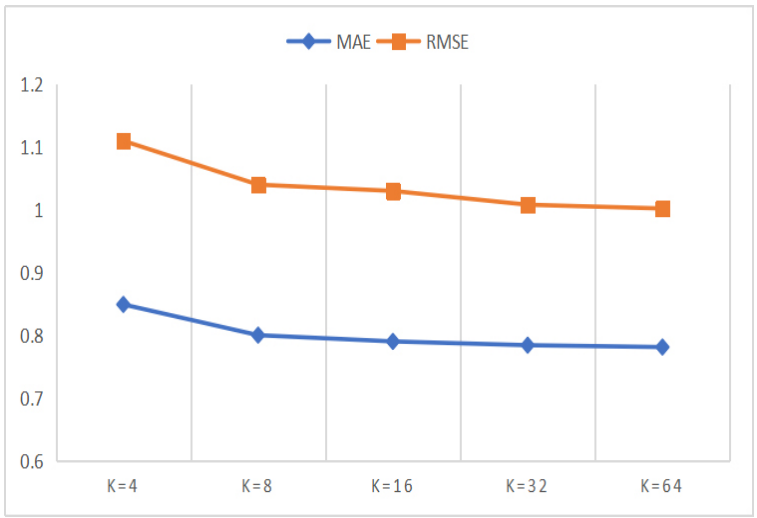

Figure 6. The performances of the DANER model with different K on Yelp dataset.

In Figures 5 and 6, we can observe that the experimental performance becomes better with the increase of the K value. However, when the K value reaches 32, the improvement speed of experimental performance tends to be slow, while the growth speed of the computational cost is more intense. Therefore, considering both the experimental performances and the computational cost of the model, we plan to set the K value to 32. At the same time, we can also observe that the MAE and RMSE values for the Amazon dataset are better than

for the Yelp dataset. This can be attributed to the fact that the Yelp dataset is more sparse than the Amazon dataset.

5.4.5. Interpretability of Recommendation Results

The interpretative recommendation results are more reasonable, more persuasive, and more capable of gaining the trust of users. In the process of the experiment, we can obtain the attention coefficients of the global attention network, which provide a basis for further studying the attention mechanism and distinguishing the critical meta-path. Moreover, we can provide an explanation based on the critical meta-path for recommendation results. Specifically, we randomly selected seven groups of user rating records and visualized their global attention coefficients, which are shown in Figures 7 and 8.

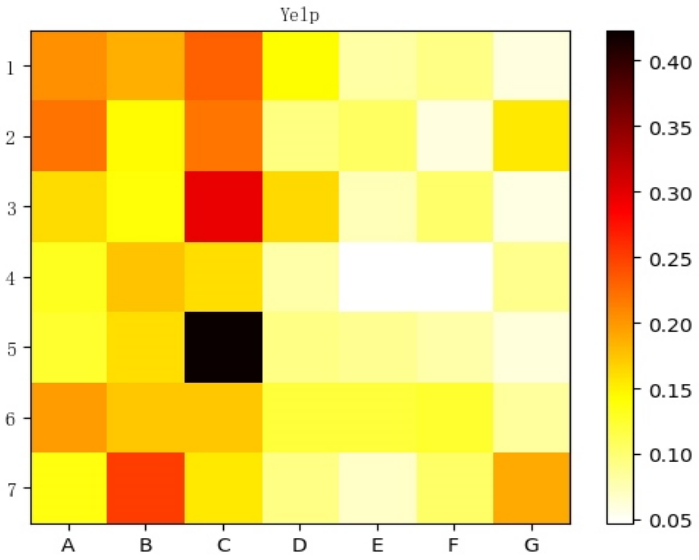

Figure 7. Attention coefficients corresponding to the seven meta-paths of seven records.

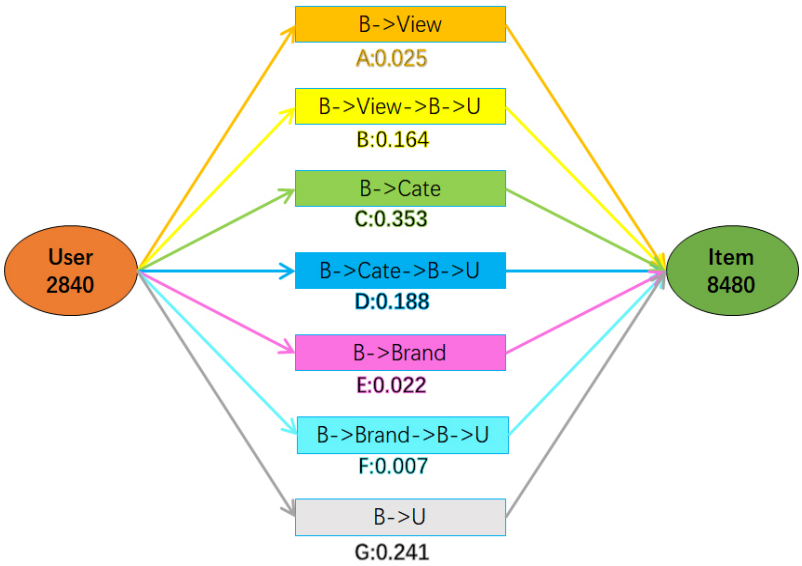

Figure 8. Attention coefficients corresponding to the seven meta-paths in record 5.

The abscissas in Figure 7 are seven predefined meta-paths, which are named *A*, *B*, *C*, *D*, *E*, *F* and *G* respectively. The ordinates are seven pairs of user and item rating records selected randomly. Each block in the figure corresponds to the attention coefficient of the corresponding meta-path in the record, and its numerical value corresponds to the color

depth. Figure 8 shows more detailed information about record 5, including the user ID, item ID, meta-path type, and the size of the attention coefficient.

As shown in Figure 8, for the record 5, the user ID is 2840, the item ID is 8480, and the predicted rating of the user for the item is 5, which indicates that user 2840 has a strong desire to buy item 8480. At the same time, the meta-path with the maximum attention coefficient in this record is C $(U \rightarrow B \rightarrow Cate \leftarrow B)$. Therefore, we can provide an explanation based on the meta-path $U \rightarrow B \rightarrow Cate \leftarrow B$, that is, the reason why we recommend item 8480 to user 2840 is that the user 2840 has purchased items with the same category as item 8480.

Based on this situation, several explanations can be specified in advance according to the meta-path. For each result of the recommendation, the explanation corresponding to the meta-path with the largest attention coefficient will be selected as the reason for recommendation.

6. Conclusions

In this paper, we proposed a rating prediction framework based on heterogeneous information networks and attention mechanisms. We exploited meta-paths to mine the high-level relationship between users and items in heterogeneous information networks. Then, we adopted a matrix decomposition to generate the latent representations of users and items. After that, we designed local and global attention neural networks to obtain the user–item joint representations integrating multiple meta-path information. By the interaction model, we can obtain the predicted ratings of users on items. The results of several experiments demonstrate that the DANER model is superior to most existing rating prediction models on achieving higher recommendation accuracy. Moreover, we visualized the attention coefficients to explain the recommendation results, which are more trustworthy.

Author Contributions: Conceptualization, Y.W. and X.H.; Methodology, T.J. and X.Z.; Formal analysis, T.J.; Resources, Y.W.; Writing—original draft, X.Z., T.J. and X.H.; Writing—review & editing, X.Z. and X.H.; Project administration, B.Y.; Funding acquisition, B.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (No. 62272191, No. 61976102, No. 61872161), the Science and Technology Development Program of Jilin Province (No. 20220201153GX), and the Interdisciplinary and Integrated Innovation of JLU (No. JLUXKJC2020207).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Amazon: https://nijianmo.github.io/amazon/index.html#files accessed on 14 November 2022; Yelp: https://www.yelp.com/dataset accessed on 14 November 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shao, B.; Li, X.; Bian, G. A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *Expert Syst. Appl.* **2021**, *165*, 113764. [CrossRef]
2. Wang, S.S.; Pan, Y.Y.; Yang, X. Research of Recommendation System Based on Deep Interest Network. *J. Phys. Conf. Ser.* **2021**, *1732*, 012015. [CrossRef]
3. Wang, W.; Feng, F.; He, X.; Nie, L.; Chua, T.S. Denoising Implicit Feedback for Recommendation. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Virtual, 8–12 March 2020. [CrossRef]
4. Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.Y. Collaborative knowledge base embedding for recommender systems. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016. [CrossRef]
5. He, X.; Chua, T.S. Neural factorization machines for sparse predictive analytics. In Proceedings of the SIGIR 2017—The 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017. [CrossRef]

6. Lu, W.; Altenbek, G. A recommendation algorithm based on fine-grained feature analysis. *Expert Syst. Appl.* **2021**, *163*, 113759. [CrossRef]
7. F, S.Y.A.; B, H.W.; C, Y.L.; D, Y.Z.; E, L.H. Attention-aware Metapath-based Network Embedding for HIN based Recommendation. *Expert Syst. Appl.* **2021**, *174*, 114601.
8. Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W.L.; Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2018. [CrossRef]
9. Geng, S.; Fu, Z.; Tan, J.; Ge, Y.; de Melo, G.; Yongfeng, Z. Path Language Modeling over Knowledge Graphs for Explainable Recommendation. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022.
10. Wang, P.; Cai, R.; Wang, H. Graph-based Extractive Explainer for Recommendations. *arXiv* **2022**, arXiv:2202.09730.
11. Zhang, J.; Yu, P.S.; Zhou, Z.H. Meta-path based multi-network collective link prediction. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014. [CrossRef]
12. Sinha, B.B.; Dhanalakshmi, R. DNN-MF: Deep neural network matrix factorization approach for filtering information in multi-criteria recommender systems. *Neural Comput. Appl.* **2022**, *34*, 10807–10821. [CrossRef]
13. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.S. Neural graph collaborative filtering. In Proceedings of the SIGIR 2019—The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019. [CrossRef]
14. Chae, D.K.; Kim, S.W.; Kang, J.S.; Lee, J.T. CFGAN: A generic collaborative filtering framework based on generative adversarial networks. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018. [CrossRef]
15. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
16. Salakhutdinov, R.; Mnih, A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20*; Neural Information Processing Systems Foundation: Vancouver, BC, Canada, 2009.
17. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI, Montreal, QC, Canada, 18–21 June 2009.
18. Rendle, S. Factorization machines. In Proceedings of the IEEE International Conference on Data Mining, ICDM, Sydney, Australia, 13–17 December 2010. [CrossRef]
19. Koren, Y. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data* **2010**, *4*, 1–24. [CrossRef]
20. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based recommendation with graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence 2019, Atlanta, GA, USA, 8–12 October 2019. [CrossRef]
21. Tran, D.H.; Sheng, Q.Z.; Zhang, W.E.; Aljubairy, A.; Zaib, M.; Hamad, S.A.; Tran, N.H.; Khoa, N.L.D. HeteGraph: Graph learning in recommender systems via graph convolutional networks. *Neural Comput. Appl.* **2021**, 1–17. [CrossRef]
22. Xiong, X.; Qiao, S.; Han, N.; Xiong, F.; Bu, Z.; Li, R.H.; Yue, K.; Yuan, G. Where to go: An effective point-of-interest recommendation framework for heterogeneous social networks. *Neurocomputing* **2020**, *373*, 56–69. [CrossRef]
23. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; Yin, D. Graph neural networks for social recommendation. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13 May 2019. [CrossRef]
24. Cao, Y.; Wang, X.; He, X.; Hu, Z.; Chua, T.S. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13 May 2019. [CrossRef]
25. Dong, Y.; Chawla, N.V.; Swami, A. Metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017. [CrossRef]
26. Fu, T.Y.; Lee, W.C.; Lei, Z. HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning. In Proceedings of the International Conference on Information and Knowledge Management, Singapore, 6–10 November 2017. [CrossRef]
27. Yu, X.; Ren, X.; Gu, Q.; Sun, Y.; Han, J. Collaborative Filtering with Entity Similarity Regularization in Heterogeneous Information Networks. In Proceedings of the IJCAI-13 HINA workshop (IJCAI-HINA'13), Beijing, China, 3–9 August 2013.
28. Yu, X.; Ren, X.; Sun, Y.; Gu, Q.; Sturt, B.; Khandelwal, U.; Norick, B.; Han, J. Personalized entity recommendation: A heterogeneous information network approach. In Proceedings of the WSDM 2014—7th ACM International Conference on Web Search and Data Mining, New York, NY, USA, 24–28 February 2014. [CrossRef]
29. Shi, C.; Zhang, Z.; Luo, P.; Yu, P.S.; Yue, Y.; Wu, B. Semantic path based personalized recommendation on weighted heterogeneous information networks. In Proceedings of the International Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015. [CrossRef]
30. Zhao, H.; Yao, Q.; Li, J.; Song, Y.; Lee, D.L. Meta-graph based recommendation fusion over heterogeneous information networks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017. [CrossRef]
31. Xie, J.; Zhu, F.; Li, X.; Huang, S.; Liu, S. Attentive Preference Personalized Recommendation with Sentence-level Explanations. *Neurocomputing* **2020**, *426*, 235–247. [CrossRef]

32. Wang, H.; Wang, N.; Yeung, D.Y. Collaborative deep learning for recommender systems. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015. [CrossRef]
33. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
34. Song, W.; Duan, Z.; Xu, Y.; Shi, C.; Zhang, M.; Xiao, Z.; Tang, J. Autoint: Automatic feature interaction learning via self-attentive neural networks. In Proceedings of the International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019. [CrossRef]
35. Chen, L.; Zheng, Z.; Liu, Y.; Yu, P.S. Heterogeneous neural attentive factorization machine for rating prediction. In Proceedings of the International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018. [CrossRef]
36. Chen, J.; Wang, X.; Zhao, S.; Qian, F.; Zhang, Y. Deep attention user-based collaborative filtering for recommendation. *Neurocomputing* **2020**, *383*, 57–68. [CrossRef]
37. Chen, J.; Zhang, H.; He, X.; Nie, L.; Liu, W.; Chua, T.S. Attentive collaborative filtering: Multimedia recommendation with item-And component-level attention. In Proceedings of the SIGIR 2017—The 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017. [CrossRef]
38. Velicković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. *STAT* **2017**, *1050*, 20.
39. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. KGAT: Knowledge graph attention network for recommendation. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2019. [CrossRef]
40. Wang, X.; Ji, H.; Cui, P.; Yu, P.; Shi, C.; Wang, B.; Ye, Y. Heterogeneous graph attention network. In Proceedings of the The Web Conference 2019—The World Wide Web Conference, San Francisco, CA, USA, 13 May 2019. [CrossRef]
41. Lu, R.; Hou, S. On semi-supervised multiple representation behavior learning. *J. Comput. Sci.* **2020**, *46*, 101111. [CrossRef]
42. Juan, Y.; Zhuang, Y.; Chin, W.S.; Lin, C.J. Field-aware factorization machines for CTR prediction. In Proceedings of the RecSys 2016—10th ACM Conference on Recommender Systems, Boston, MA, USA, 17 September 2016. [CrossRef]
43. Guo, H.; Tang, R.; Ye, Y.; Li, Z.; He, X. DeepFM: A factorization-machine based neural network for CTR prediction. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017. [CrossRef]
44. Cheng, H.T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & deep learning for recommender systems. In Proceedings of the ACM International Conference Proceeding Series, Indianapolis, IN, USA, 24–28 October 2016. [CrossRef]
45. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the 26th International World Wide Web Conference, Perth, Australia, 3–7 April 2007. [CrossRef]
46. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.