Temporal Meta-path Guided Explainable Recommendation

Hongxu Chen
University of Technology Sydney
Hongxu.Chen@uts.edu.au

Yicong Li*
University of Technology Sydney
Yicong.Li@student.uts.edu.au

Xiangguo Sun
Southeast University
sunxiangguo@seu.edu.cn

Guandong Xu†
University of Technology Sydney
Guandong.Xu@uts.edu.au

Hongzhi Yin*
The University of Queensland
h.yin1@uq.edu.au

ABSTRACT

Recent advances in path-based explainable recommendation systems have attracted increasing attention thanks to the rich information provided by knowledge graphs. Most existing explainable recommendation only utilizes static knowledge graph and ignores the dynamic user-item evolutions, leading to less convincing and inaccurate explanations. Although there are some works that realize that modelling user's temporal sequential behaviour could boost the performance and explainability of the recommender systems, most of them either only focus on modelling user's sequential interactions within a path or independently and separately of the recommendation mechanism. In this paper, we propose a novel **T**emporal **M**eta-path Guided **E**xplainable **R**ecommendation (**TMER**), which utilizes well-designed item-item path modelling between consecutive items with attention mechanisms to sequentially model dynamic user-item evolutions on dynamic knowledge graph for explainable recommendations. Compared with existing works that use heavy recurrent neural networks to model temporal information, we propose simple but effective neural networks to capture users' historical item features and path-based context to characterise next purchased item. Extensive evaluations of TMER on three real-world benchmark datasets show state-of-the-art performance compared against recent strong baselines.[1]

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

explainable recommendation, temporal recommendation

---

*Having equal contribution with the first author
†Corresponding author
[1]https://github.com/Abigale001/TMER

---

1 INTRODUCTION

Reasoning with paths over user-item associated Knowledge Graphs (KGs) has been becoming a popular means for explainable recommendations [32, 34, 44]. The path-based recommendation systems have successfully achieved promising recommendation performance, as well as appealing explanations via searching the connectivity information between users and items across KGs. One category of existing works on path-based explainable recommendations seeks auxiliary meta-paths (pre-defined higher-order relational compositions between various types of entities in KGs) as similarity measures and evidence for possible explanations between users and items.
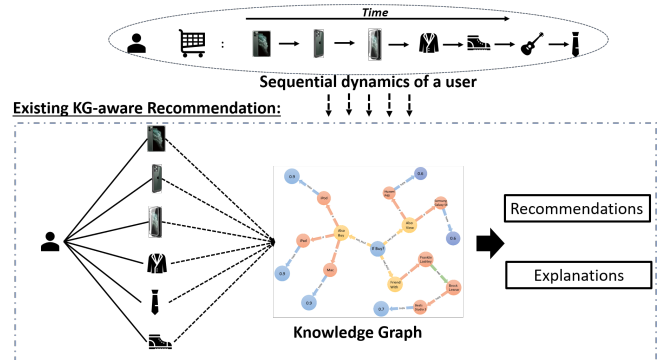

Figure 1: The concept of TEMR. TEMR contributes a sequential dynamic modelling layer on top of existing knowledge-aware explainable recommendation (the dashed box on the bottom).

However, most of existing path-based methods for explainable recommendation simply treat the underlying KGs as static graphs, ignoring the dynamic and evolving nature of user-item interactions in real-world recommendation scenarios. The dynamics and evolutions of users' interactions with items play a pivotal role in both recommendation precision and explanations for a users' real intent. Take the scenario in Figure 1 as an example, a user purchased a phone case and a phone film after a recent buy of a new phone. If we ignore the sequential information between each purchase and treat the whole information as a static graph, the system probably gives an explanation of buying the phone case is because a similar customer also bought this phone case by exploring co-purchasing

relationships. Whilst the explanation, in this case, might be valid and the underlying reasoning mechanism (collaborative filtering signal) can be used for recommendation, it is still sub-optimal as a more appealing motivation of buying a phone case is due to the recent purchase of a new phone. For this reason, in this example, it is important for a system to be capable of considering temporal and sequential user-item interactions and disentangling the importance of various reasons when generating possible explanations.

Although some prior works have considered some extent of the sequential information for knowledge-aware path-based explainable recommendation problem, they still fail to explicitly model the dynamics of users' activities. To allow effective reasoning on paths to infer the underlying rationale of a user-item interaction, the method proposed in [32] takes the advantages of path connectivity and leverages the sequential dependencies of entities and sophisticated relations of a path connecting a user-item pair. Nevertheless, the methods only consider the sequential information within specific paths and did not consider the importance of the user' historical sequences on reflecting the user's dynamic interactions with items. To improve, the KARN model [45] fuses the user's clicked history sequence and path connectivity between users and items in KGs for recommendation. However, the method models the user's sequential behaviours and user-item interactions separately and in a coarse-grained manner (treating a user's click history as a whole), which may restrict the expressiveness of users' temporal dynamics on recommendation explainability.

In light of this, in this paper, we challenge the problem of exploring users' temporal sequential dynamics in the context of path-based knowledge-aware recommendation. Different from existing works that either only consider the sequential information within a path or treat the user's sequential interactions as a whole and separately, we aim to 1) explicitly model and integrate the dynamic user-item interactions over time into the path-based knowledge-aware recommendation, and 2) leverage the captured dynamics of user-item interactions to improve the performance and explainability of the recommendation .

It is worth noting that modelling users' temporal sequential behaviour with path-based knowledge-aware explainable recommendation is a non-trivial and challenging task. First, path-based knowledge-aware recommender systems are built upon the well-constructed KGs, and its expected accuracy and explainability are highly related to the underlying KGs and distilled paths. If temporal information between users and items is considered, the original underlying static KGs will be cast into multiple snapshots w.r.t. the timestamps. Comparing with the static KGs that consist of all users' full timelines, each snapshot at a certain timestamp only has partial observations of the global knowledge, which will result in inferior recommendation performance. To deal with the issue, we devise a general framework for temporal knowledge aware path-based explainable recommender systems, namely **Temporal Meta-path Guided Explainable Recommendation (TMER)**. TMER provides a solution that can explicitly depict users' sequential behaviour while being able to be aware of global knowledge of the entire underlying KGs. Specifically, to model the temporal dynamics between users and items, TMER naturally models the task as a sequential recommendation problem and takes as input users and their corresponding sequential purchase history. To learn the

sequential dependencies between consecutive items purchased by a user, TMER novelly explores item-item paths between consecutive items and embed the paths as context with elaborated designed attention mechanisms to model the dynamics between user and items. Compared to existing path-based explainable recommendation systems that only consider user-item paths as evidence and support for the recommendation decision, TMER contributes another creamy layer on the top of existing works, which makes use of the powerful expressiveness of temporal information between users and items.

In addition, when taking paths into consideration for the above purpose, another challenge lies in the existence of a large number of possible paths. It is time-consuming for a model to select several paths that are meaningful, expressive, and have positive impacts on both recommendation performance and explainability. To this end, inspired by prior work [16], we leverage the concept of meta-path [27] and explore diverse meta-path schemas to characterize the context of dynamic interactions between users and items. With resort to the powerful transformer model for sequential modelling, we also elaborate item-item path attention and user-item path attention units to learn combinational features of multiple paths to further characterize users' temporal purchasing motivations and their general shopping tastes, respectively. The rationale for developing such path-based attention units is that a user's motivation towards buying a certain product is complex and consists of multiple factors. For example, when buying a new phone, a customer may consider several factors including a phone's intrinsic features such as functionality, display, camera, etc., as well as other external factors such as the choices of their close friends, and their previous purchase of certain related electronics using the same operating system. With help of the designed path-based attention units, the proposed TMER framework is able to learn different weights for various possible paths which will be then used as explanations for recommendations (we show the effectiveness of using the proposed path-based attention units for the explainable recommendation in the experiments by running case studies).

Another shining point of our work is that the introduced item-item path modelling also serves as the core of the simple yet effective sequential modelling architecture of TMER. The combinational meta-path enriched features learned by the item-item path attention units are aggregated with features of previous purchased item will serve as the prediction signals for the next item. The intuition behind this framework is that the item-item paths connect two consecutive items purchased by a user, and represent various reasons and factors that may lead to the next purchased item, which intrinsically provides strong sequential dependencies between items. Compared to existing works on the sequential recommendations that rely on Recurrent Neural Nets such as (RNNs [19], GRUs [14], LSTM [32, 45]), the proposed methods in this paper is light, simple but effective.

The main contributions of this paper are summarized as follows:

- We point out that explicitly modelling dynamic user-item interactions over time can significantly benefit the recommendation performance and explainability.
- We propose **T**emporal **M**eta-path Guided **E**xplainable **R**ecommendation (**TMER**), which considers users' dynamic

2

behaviours on top of the global knowledge graph for sequential-aware recommendation and explores both user-item and item-item meta-path paths with well-designed attention mechanisms for explainable recommendation.

- Extensive evaluations on three real-world datasets have been conducted, and the experimental results showcase the superiority, effectiveness and temporal explainability of the proposed TMER model.

2 PRELIMINARIES

We first give some essential definitions and define the problem.

Definition 2.1. **Information Networks.** An information network is a simple graph $G = (\mathcal{V}, E)$. Each edge $e \in E$ represents a particular relation $r \in R$ of two entities (v_1, v_2) linked by edge e. Each entity $v \in \mathcal{V}$ belongs to a particular type T. (v_1, r, v_2) is a triplet means head entity v_1 to tail entity v_2 is with relation r. In general, relation r and its reverse r^{-1} is not the same unless the relation is symmetrical.

Definition 2.2. **Heterogeneous Information Network.** A heterogeneous information network (HIN) is a special type of information network. In heterogeneous information network H, edges E and entities $\mathcal{V}$ are in different types, that is, $|R| > 1, |T| > 1$.
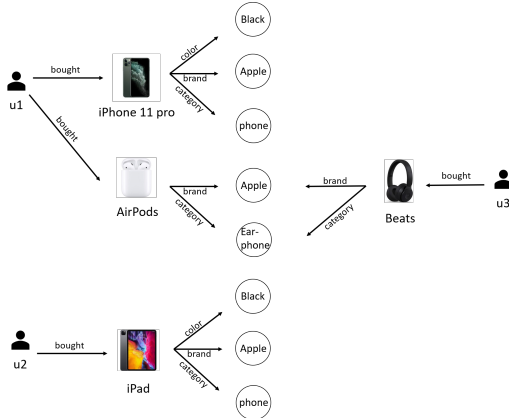

Figure 2: An example of a heterogeneous information network: product recommendation network

EXAMPLE 1. *As shown in Figure 2, A product recommendation network G_P is a HIN, in which entities $\mathcal{V}$ could be many types, including items, users, categories, colors, brands, etc. E could represent different relations. Edges between users and items denote the buy relation, edges between items and brands denote the is brand of relation, etc.*

Definition 2.3. **Meta-path**. A meta-path [27] P in a network from entity v_0 to v_k, is denoted as $v_0 \xrightarrow{r_0} v_1 \xrightarrow{r_1} v_2 \xrightarrow{r_2} ... \xrightarrow{r_{k-1}} v_k$, where composite relation from v_0 to v_k is $r = r_0 \circ r_1 \circ r_2 \circ ... \circ r_{k-1}$. $\circ$ represents the composition operator on relations.

EXAMPLE 2. *In the product recommendation network G_P in figure 2, there are many meta-paths; one of the meta-paths is UIBI, which means user $\rightarrow$ item $\rightarrow$ brand $\rightarrow$ item. In our paper, the chosen meta-paths will be discussed in the later sections.*

Symbol	Description
G	Information network
H	Heterogeneous information network (HIN)
$e \in E$	an edge
$v \in \mathcal{V}$	an entity or node
R	Type of edges (relations)
T	Type of nodes (entities)
$p \in P$	a meta-path in a HIN
$u \in U$	User u
$i \in I$	Item i
W	Weight matrix
$\mathbf{b}$	Bias vector
$\phi \in \Phi$	path instance in a HIN
h	Embedding
α	Attention mechanism parameter
$\hat{r}$	User-item rating

Table 1: Description of major notations used in this paper

PROBLEM 1. ***Sequential knowledge-aware explainable recommendation.*** *For a user $u_i \in U$, given the item set I, the user transaction sequence π_u and the item associated information network G, the target of knowledge-aware explainable recommendation is to predict top k items that u_i will interact with, as well as the possible reasonings of recommended items.*

3 MODEL

In this section, we introduce the proposed model Temporal Meta-path Guided Explainable Recommendation. In the remaining of the paper, we use the notation summarized in Table 1 to refer to the variables and parameters used throughout the paper.

3.1 Overview of TMER Architecture

The overall architecture of the proposed TMER model is shown in Figure 3. It mainly consists of four components. First, to initialize users and items, we use DeepWalk [24] to embed user and item entities. Secondly, instead of simply utilizing meta-paths, we mine and extract both meaningful sequential (temporal) and non-sequential meta-paths instances to improve the recommendation performance and personalization. In this step, we obtain item-item instance paths between consecutive items. After embedding instances and user/item vectors with attention mechanism, we consider the weights of instances as the weights of reasoning paths for the specific user. We utilize different weights of reasoning paths to update item embeddings making items contain reasoning information. This step also models the users' sequential purchased information, feeding the previous item's feature to the next one. Finally, we feed item embeddings, user embeddings, and instances to the recommendation network to do recommendation. The specific steps are elaborated in the following subsections.

3.2 Initialize User and Item Representations

We firstly learn latent representations for involved users and items by treating random truncated random walks in a user-item bipartite network as an equivalent of sentences in DeepWalk [24], which optimizes the co-occurrence probability among the entities in a
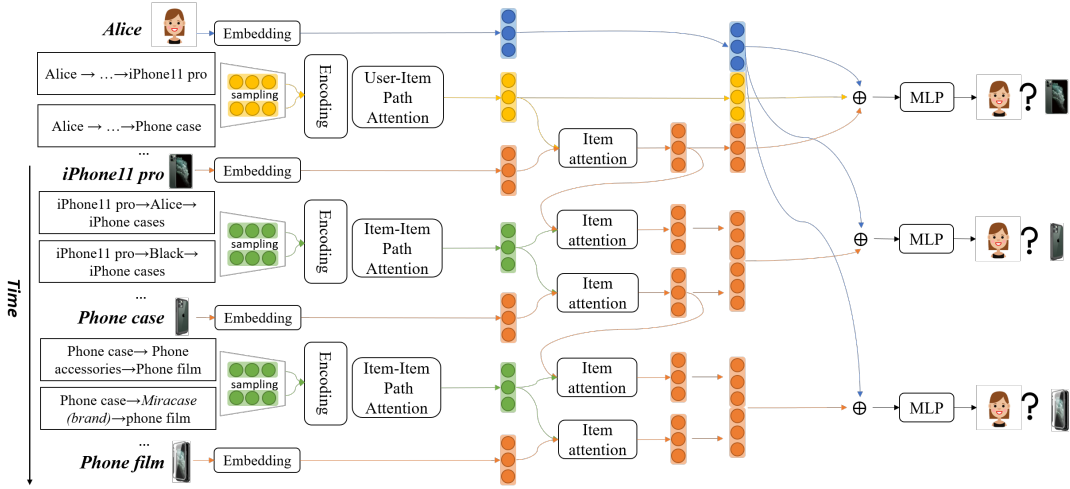
Figure 3: The architecure of Temporal Meta-path Guided Explainable Recommendation. Here shows an example: User _Alice_ bought _iPhone 11 pro, Phone case_ and _Phone film_ in a sequential order and the training process for _Alice._

walk by using skipgram based on word2vec [22]. Other recent advanced graph-based embedding initialization methods can be also applied, like PME [4], GraphSage [9], GAT [30], and so on. These recent works usually can even outperform DeepWalk. However, through extensive comparison with these methods, DeepWalk is the best choice, the possible reason behind this is because DeepWalk pays more attention to the embedding of nodes in a path, while GCNs, such as GraphSage, GAT, etc., learn embedding of each entity with the aggregated feature information from its local neighborhoods. That is, these approaches focus on modelling local relationships. However, if only considering the recommendation task itself, although the local relationships are significant, the global higher-order relationships learned by walks on a graph also play a notable role. For example, the transitivity of co-purchasing relationships between friends and the substitute relationship of items can be discovered by modelling higher-order relations in a bipartite user-item graph.

3.3 Incorporating Meta-path based Context

In recommendation tasks, external features related to users and items, such as product attributes [5, 37], user social information [3, 8, 38], and user demographic information are usually considered as additional auxiliary information to complement traditional recommendation methods. However, how to utilize the heterogenous additional information efficiently is an open problem. Some prior works [18, 20, 41] attempted to consider social relations as the user-side information to boost the recommendation performance. To seek help from injecting more complex additional information, recent works [10, 40] introduced meta-paths into recommendation methods to describe relational compositions between various types of entities in heterogeneous information networks. In [40], the authors proposed to diffuse user preferences along different meta-paths in information networks to generate latent features of users and items. Related work [10] firstly extracts different-aspect features with meta-paths from a HIN, and then fuse aspect-level latent factors to the recommendation systems. However, these methods largely rely on the latent factors obtained from constructed meta-path based similarity matrix, which are too general and only

can reflect mutual interaction between different types of entities in a graph but cannot capture the specific information along particular path instances. Therefore, inspired by existing work [16], we explore to improve both recommendation performance and explainability by modelling more specific meta-path instances.

Different from existing works, we differentiate meta-path instances into two different categories based on the involved entities in a recommendation scenario (i.e., user-item and item-item meta-path instances). Through modelling these path instances, we learn a more detailed meta-path based context to further characterize the motivations, reasons as well as leading factors between each pair of user-item interaction. While previous works such as [16] mainly focuses on modelling meta-path instances between a user and an item, this paper highlights the item-item meta-path instances, which we think is beneficial in multiple aspects to sequential explainable recommender systems. Firstly, only considering user-item paths is restrictive for recommendation explainability as user-item paths only represent a user's general shopping interests. In comparison, item-item paths are more expressive and can reflect devise reasons by exploring higher-order relations among items, such as complementary products (e.g., phone and a phone case), substitutable items of known items (e.g., iPhone - phone film - Huawei Phone), co-purchased products with other people, etc. In addition, item-item paths sometimes also serve as sequential modelling signals that naturally capture the temporal dependencies between each consecutive item purchased by users, which will be of great impact for sequential explainable recommendations.

Despite the powerful expressiveness of meta-paths in exploring HIN based knowledge-aware recommendation, it is still challenging mainly because the number of meta-paths is too large to handle (i.e., the amount of edges is cubic to the number of entities). Taking the electric product recommendation situation for an example, for $IUIBI$ meta-path schema, if we fix the starting node - iPhone11 pro, there are many instances: $iPhone11\ pro \rightarrow Alice \rightarrow iPad \rightarrow Apple \rightarrow AirPods$, $iPhone11\ pro \rightarrow Amy \rightarrow Phone\ case \rightarrow Miracase \rightarrow Phone\ film$, and so on. Therefore, it is necessary to sample useful path instances while limiting the total

amount to simplify later calculations. Intuitively, if the embedding of the next node and the current node is similar, the path to the next node is more likely to be connected. Therefore, according to this assumption, we use a method based on adjacent nodes' similarity to sample the next node along the path. Specifically, for each specified meta-path schema, we measure the priority degree by calculating the similarity between the current node and candidate out-going nodes. Such a priority score directly reflects the association degree between two nodes. Through this approach, we sample and get a few meta-path instances for each user-item pair and item-item pair to participate in recommendation module training. Taking meta-path $UIBI$ as an example, we firstly sample meta-path instances according to the similarity of U and I, and B and I to get the start sub-paths and end sub-paths, getting top k similar paths, and then, we calculate the similarity of I and B to get top k middle paths to get all sampled paths. In what follows, we elaborate on how to embed the path-based context.

3.4 Parameterizing Combinational Features of Meta-paths as Recommendation Context

*3.4.1 **Learning combinational path-based features with Self-Attention**.* After obtaining sampled user-item meta-path instances and item-item meta-path instances, we first regard paths as sentences, nodes as tokens in sentences, using Word2Vec [22] method and $Mean(\cdot)$ operations to learn path embedding. Then, we employ multi-head self-attention units to learn the meta-path based context (the User-Item and Item-Item Path Attention modules shown in Figure 3). The rationale of deploying such self-attention units here is that because, after sampling, there are still multiple paths between each pair of item-item (or user-item) representing particular distinct reasons (reasoning paths); and we observe that the reasons for buying consecutive two items are not simply unique; rather, the reasons are more complex and likely a mixture of multiple different reasons. For example, the reasons for a customer to buy a phone case right after his/her previous purchase of a new phone are probably a mixture of 1) his/her friends who own a similar phone and bought this particular phone case, 2) the phone case is the most popular match for the purchased new phone, 3) the color of the phone case matches the customer's preference. The potential reasons can be more than the listed, and again they can be represented by using various meta-paths.

Based on this observation, self-attentive properties of the Transformer model [29], we aim to learn the combinational features from multiple path instances to better characterize the complex reasons between each connected pair of entities in the KG.

$$Attention(Q_\phi, K_\phi, V_\phi) = Softmax(\frac{Q_\phi K_\phi^T}{\sqrt{d_k}})V_\phi,$$

$$MultiHead(Q_\phi, K_\phi, V_\phi) = Concat(head_1, ..., head_m)W^O,$$

(1)

where $head_i$ is $Attention(W_i^Q Q_\phi, W_i^K K_\phi, W_i^V V_\phi)x$. Query Q, key K and value V are self-attention variables associated with path ϕ, and W is the weight. d_k is the dimensionality (here $d_k = 100$). $Concat(.)$ is the concatenation operation.

*3.4.2 **Modelling temporal dependencies with item-item meta-path instances**.* To learn the temporal dynamics of each

user, the proposed TMER framework resorts to the above-mentioned item-item meta-path instances together with the well-designed architecture to capture the sequential dependencies between two consecutive items. Compared with most existing works on sequential recommendation [32, 35, 45] that utilize recurrent neural networks to encode the temporal effects between items in a user's interacted sequence, the proposed TMER bypasses the de-facto default deployment of RNNs or LSTMs that sometimes make the model even heavier. Specifically, the proposed framework novelly model the temporal dependencies between two items by capturing 1) the information passed from the previous item through an item-attention unit, 2) item-item connectivity through a specific sampled path instance. Notably, the information passed from the previous item is an attentive aggregation of previous item-item connectivity information. For example, in Figure 3, whether Alice will buy the phone film is modelled by considering 1) the information passed from the phone case (which includes the connectivity between iphone11 pro and the phone case), and 2) the paths between the phone case to the phone film. As a result, the long-range and short-term "memory" in a sequence can be captured, and the extent of the goodness of the long and short term memory can be influenced by the length of the modelled sequence in different scenarios with different datasets.

*3.4.3 **Updating item representations**.* After updating representations of user-item meta-path instances and item-item meta-path instances according to a multi-head self-attention mechanism, we employ a different kind of attention mechanism to update item representations. It is obvious that the current item is mostly related to the last one, which means it is better to add the last item's information to the current one to contain temporal information. Besides, the current item is also related to the instances that arrived at this item. Therefore, we perform a two-layer attention mechanism to update item representations. Mathematically,

$$\mathbf{h}_i^{(1)} = ReLU(W_{i-1}\mathbf{h}_{i-1} + W_{\phi_{i-1 \to i}}^{(1)} \mathbf{h}_{\phi_{i-1 \to i}} + \mathbf{b}_i^{(1)}) \odot \mathbf{h}_{i-1} \quad (2)$$

$$\mathbf{h}_i^{(2)} = ReLU(W_i\mathbf{h}_i + W_{\phi_{i-1 \to i}}^{(2)} \mathbf{h}_{\phi_{i-1 \to i}} + \mathbf{b}_i^{(2)}) \odot \mathbf{h}_i \quad (3)$$

where $\mathbf{h}_i^{(1)}$ and $\mathbf{h}_i^{(2)}$ mean the first and second layer output of the item attention module, respectively. $\mathbf{h}_{i-1}$ is the last item's latent representation. $\phi_{i-1 \to i}$ is the instance from the $(i-1)^{th}$ item to the ith item. W and $\mathbf{b}$ with different superscripts denote different variables' weights and bias. However, until now, here is a problem with the calculation of the first item, because there is no item before it. To solve this problem, we involve the user-item instance from user u to the first item into the update of the first item, as Eq. 4 shows. Actually, the instance from u to the first item is really important in the recommendation, because it is the first item user has bought and most of the other bought items have a sequential relation with the first item to some extent. That is why we embed it into the first item.

$$\mathbf{h}_{i=1} = ReLU(W_i\mathbf{h}_i + W_{\phi_{u \to i}}\mathbf{h}_{\phi_{u \to i}} + \mathbf{b}_i) \odot \mathbf{h}_i \quad (4)$$

where $\phi_{u \to i}$ represents the path from user u to the first item.

Table 2: Dataset Information.

Datasets	User	Item	Brand	Category
Musical Instruments	1450	11457	1185	429
Automotive	4600	36663	3790	1592
Toys and Games	9300	58743	5404	820

3.5 The Complete Recommendation Model

Finally, we concatenate user, item and instances information (calculated in 3.4.2) to a vector according to Eq. 5, and get user-item prediction scores through Multilayer Perceptron (MLP) with explainability instances.

$$h_{u,i} = [\mathbf{h}_u; \mathbf{h}_i^{(1)}; \mathbf{h}_i^{(2)}] \tag{5}$$

where $[;]$ means vector concatenation. Here $h_{u,i}$ denotes the explicit mutual vector of the user, item, and implicit mutual of user-item meta-path instances and item-item meta-path instances. For the first item, the concatenation operation is different because of the dimension problem, and therefore, for the first item related to each user, the vector fed in MLP involves user-item instance, mathematically,

$$h_{u,i=1} = [\mathbf{h}_u; \mathbf{h}_{\phi_{u \to 1}}; \mathbf{h}_{i=1}] \tag{6}$$

After that, final user-item rating calculates as follows.

$$r_{u,i} = MLP(h_{u,i}) \tag{7}$$

where the MLP contains a two-hidden-layer neural network with ReLU as the activation function and sigmoid function as the output layer. According to [11, 16], neural network models can learn more abstractive features of data via using a small number of hidden units for higher layers, we employ a tower structure for the MLP component, halving the layer size for each successive higher layer.

We use implicit feedback loss with negative sampling [13, 28] as the whole loss function:

$$loss_{u,i} = -E_{j \sim P_{neg}} \left[\log \left(1 - r_{u,j} \right) \right] \tag{8}$$

where models the negative feedback drawn from the noise distribution P_{neg}, which is a uniform distribution following [16].

4 EXPERIMENT

4.1 Experiment Settings

4.1.1 Datasets. We use three widely used datasets from the Amazon[2] platform [23], including musical instruments, automotive, and toys and games dataset. Each dataset includes brand and category as the metadata. More details are shown in Table 2.

We select the latest twelve items for each user and order these items by timestamps, and then we choose the first two items as bridge items, the next four items as training items and the rest as test items. We create the Heterogeneous Information Network using *user*, *item*, *brand* and *category* in datasets.At last, user-item meta-path instances and item-item meta-path instances are generated and sampled according to section 3.4.

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

4.1.2 Evaluations. We leverage Top K Hit Ratio (HR@K) and Top K Normalized Discounted Cumulative Gain (NDCG@K) as our metrics. For each positive test instance (u, i^+), we mix the ground truth item i^+ with $\mathcal{N}$ random negative items, then rank all these $\mathcal{N}+1$ items and compute the HR@K and NDCG@K scores. We set $\mathcal{N} = 500$ and $K = 1, 5, 10, 20$ for evaluations. For evaluation of the explainability of the recommendation, we use case studies to show the explanations in detail.

4.1.3 Baselines. We briefly introduce the baseline methods for comparison below.

- **GRU4Rec**[14, 15]: GRU4Rec is a session-based recommendation method using GRU. For each user, we treat the training items as a session.
- **NARRE**[2]: NARRE utilizes neural attention mechanism to build an explainability recommendation system.
- **MCRec**[16]: MCRec develops a deep neural network with the co-attention mechanism to learn rich meta-path based context information for recommendations.
- **NFM**[12]: NFM effectively combines the linear Factorization Machines (FM) and nonlinear neural networks for prediction under sparse settings.
- **FMG**[43]: FMG uses FM and matrix factorization (MF) for recommendation and outperformed state-of-the-art FM and other HIN-based recommendation algorithms.

4.2 Parameter Settings

We choose the first second item of each user as the bridging item, the last 6 items of each user as the testing positive items, and the remaining 4 are training items. The hyperparameters are carefully tuned to achieve optimal results in all experiments. We implement GRU4Rec, NARRE and NFM based on their papers. The meta-paths and settings in MCRec are the same as the original paper. The meta-paths in FMG are *User-Item (UI)*, *User-Item-Brand-Item (UIBI)* and *User-Item-Category-Item (UICI)*. For our proposed TMER, the learning rate for Amazon Musical Instruments dataset is $5e-6$, for the Amazon Automotive dataset is $5e-5$ and for Amazon Toys and Games is $1e-4$. The parameters for other baselines are searched for their best results. We use *UIBI, UICI, UIBICI* and *UICIBI* as user-item meta-paths and *ICIBI, IBICI, ICICI, IBIBI, IUIUI, ICIUI* and *IBIUI* as item-item meta-paths. In these *user/item/brand/category* related datasets, the selected meta paths are almost all the existing 4 to 6-hop meta paths.

4.3 Effectiveness Analysis on Recommendation Results

As shown in Table 3, compared with these existing state-of-the-art recommendations, our proposed TMER achieves the best scores. In terms of the hit ratio, on toys and games dataset, we improve at least 1%, 8%, 6%, 6% to the second when $K = \{1, 5, 10, 20\}$, respectively. For NDCG, we outperform 0.9%, 3%, 6%, 3% to the second when $K = \{1, 5, 10, 20\}$, respectively. The advantages also hold on other datasets. These results demonstrate that our framework can achieve obvious advantages through explicitly modelling each user's sequential purchased information with the temporal paths, while GRU4Rec [14, 15] ignores the relation along paths and only utilizes RNN to model session-based recommendation. NARRE [2], NFM [12] and FMG [43] ignore the sequential information for each

Table 3: Performance Comparison with Baselines.

Datasets	Metrics	GRU4Rec	NARRE	MCRec	NFM	FMG	TMER
Amazon Musical Instruments	HR@1	0.6493	0.6324	0.5922	0.3929	0.4174	**0.8467**
	HR@5	0.6493	0.6408	0.6289	0.4169	0.4328	**0.9507**
	HR@10	0.6502	0.6764	0.6522	0.4631	0.6466	**0.9739**
	HR@20	0.7647	0.7988	0.6798	0.5484	0.8799	**0.9865**
	NDCG@1	0.0490	0.1025	0.1584	0.0549	0.1003	**0.2165**
	NDCG@5	0.0596	0.1224	0.1590	0.0736	0.1221	**0.2316**
	NDCG@10	0.0632	0.1572	0.1590	0.1013	0.1221	**0.2432**
	NDCG@20	0.0664	0.1603	0.1592	0.1432	0.1597	**0.2614**
Amazon Automotive	HR@1	0.6537	0.6633	0.6360	0.3362	0.4032	**0.9072**
	HR@5	0.6537	0.7063	0.6385	0.4063	0.4645	**0.9634**
	HR@10	0.6576	0.7064	0.6412	0.6024	0.8324	**0.9697**
	HR@20	0.8484	0.7963	0.6487	0.6203	0.9596	**0.9751**
	NDCG@1	0.0440	0.3343	0.1984	0.1549	0.3242	**0.4064**
	NDCG@5	0.0642	0.4024	0.2080	0.2742	0.3241	**0.4691**
	NDCG@10	0.0743	0.4536	0.2138	0.3527	0.3360	**0.4953**
	NDCG@20	0.0773	0.4761	0.2218	0.3533	0.3897	**0.5253**
Amazon Toys and Games	HR@1	0.7776	0.8235	0.6580	0.3563	0.4562	**0.8339**
	HR@5	0.7777	0.8644	0.6619	0.4794	0.5536	**0.9442**
	HR@10	0.7792	0.8933	0.6668	0.7335	0.8704	**0.9580**
	HR@20	0.8882	0.9025	0.6754	0.7544	0.8978	**0.9662**
	NDCG@1	0.1863	0.2065	0.1709	0.0954	0.1124	**0.2154**
	NDCG@5	0.2046	0.2564	0.1800	0.1543	0.1238	**0.2901**
	NDCG@10	0.2947	0.2933	0.1855	0.2566	0.1462	**0.3274**
	NDCG@20	0.2947	0.3364	0.1928	0.2594	0.1514	**0.3711**

Table 4: Impact of User-Item and Item-Item Instances Self-Attention Mechanisms.

Components	Metrics	TMER	TMER-RUI	TMER-RII
Amazon Musical Instruments	HR@1	**0.8733**	0.7844	0.7757
	HR@5	**0.9541**	0.8518	0.8374
	HR@10	**0.9742**	0.8606	0.8499
	HR@20	**0.9871**	0.8733	0.8641
	NDCG@1	**0.1688**	0.0937	0.0947
	NDCG@5	**0.2128**	0.1593	0.1500
	NDCG@10	**0.2418**	0.1938	0.1778
	NDCG@20	**0.2817**	0.2393	0.2147
Amazon Automotive	HR@1	**0.9072**	0.9060	0.8942
	HR@5	**0.9634**	0.9633	0.9421
	HR@10	**0.9697**	0.9693	0.9474
	HR@20	**0.9751**	0.9748	0.9531
	NDCG@1	**0.4064**	0.3521	0.3694
	NDCG@5	**0.4691**	0.4456	0.4472
	NDCG@10	**0.4953**	0.4794	0.4770
	NDCG@20	**0.5253**	0.5155	0.5104

user. MCRec [16] ignores the item-item intrinsic relations and just utilizes user-item interactions to train the recommendation.

4.4 Effectiveness Analysis on Attention

To study the impact of user-item and item-item instances, we further compare our model with two variants, namely TMER-RUI, and TMER-RII. TMER-RUI means we do not consider user-item instance and remove the corresponding self-attention module. TMER-RII means we remove the self-attention module for item-item instances. We report the compared results in Table 4, from which we can see that the user-item and item-item instances self-attention modules significantly boost the recommendation method, adaptively adjust the importances of different instance paths. The self-learned user-item and item-item instance paths contribute to improving

the performance of recommendation finally. Especially, using the item-item meta-path instances self-attention module helps TMER improve 6.7% and 12.3% on Musical Instruments dataset on HR@20 and NDCG@20, respectively.

4.5 Effectiveness Analysis on meta-paths

To analyze the contributions of different meta-path schemes, we randomly sample 100 negative items for each user-item positive pair and the evaluation metrics are HR@10 and NDCG@10. In Figure 5, the four sets of bars denote results of all meta-paths, meta-paths only related to users and items, meta-paths only related to users, items and brands, meta-paths only related to the users, items and categories, respectively. We find the blue and orange bars (HR@10 on Musical Instruments and Automotive datasets) are almost the same. However, we can get the conclusion from the gray and yellow bars (NDCG@10 on Musical Instruments and Automotive datasets) that meta-path related to all attributes outperform the others. Moreover, from the trend of gray or orange bars, we can find that the meta-path related to *category* improves more than *brand*.

4.6 Case Study: generating explanation for recommendation

One contribution of this paper is to give explanations on instances paths while recommending preferable items. This is because our method generates multiple item-item instance paths according to user purchase history, and then it utilizes attention mechanism to learn the weight of each item-item instance path and aggregate multiple item-item instance paths for each item-item pair. To demonstrate this, we show a random example drawn from TMER on Amazon Musical Instruments dataset.
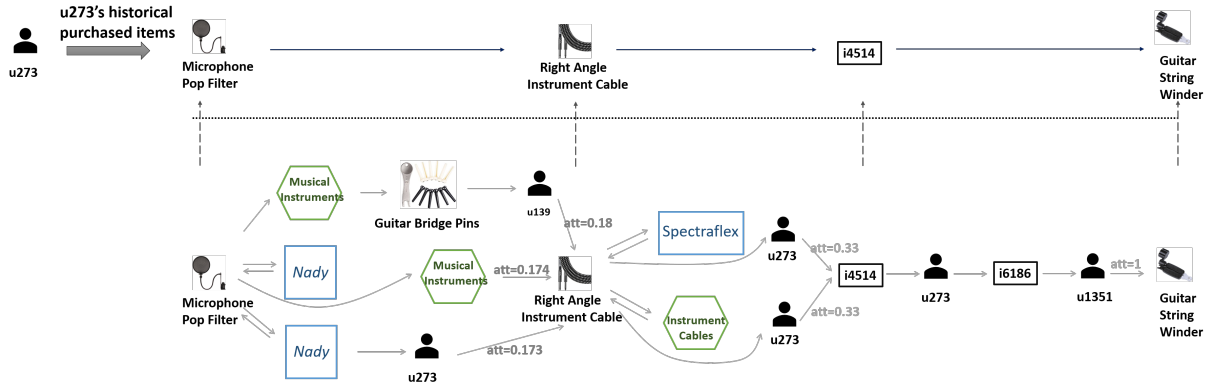
Figure 4: It shows $u273$ and related historical purchased items in the upper part of the figure. In the lower part, it displays the reasoning item-item paths along this historical purchased items related to $u273$. Green blocks represent the category attributes. Blue blocks represent the brand attributes. Black blocks without physical pictures do not have meta information in the dataset.
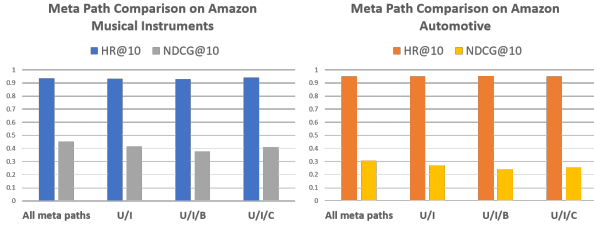

Figure 5: Impacts of different meta-paths on TMER. The x-axis and y-axis represent different meta-path types and the value of evaluation metrics, respectively.

We randomly select a user, whose id is $u273$, and track his item-item instance paths' parameters. In the training dataset, $u273$'s purchase history is $i2954$, $i2280$, $i4514$ and $i11158$. We shows several sampled item-item instance paths with high attention parameters in Figure 4 and demonstrates our explanations.

- According to Figure 4, there are three reasons for purchasing $i2280$. The most probable reason with the highest attention weight is that $u273$ bought the last musical instrument category $i2954$ and $i2237$ with the attention weight 0.18. For the next item $i4514$, the reason for purchasing it is that the user $u273$ has bought 2280 who has the same brand and category with item $i4514$. There is only one item-item instance path between some items because the item-brand and item-category data are sparse. Thus, our method can infer the reasons through the weights on different item-item paths.

- Besides, our model can capture sequential information according to user purchase history thanks to item-item paths. These item-item paths learn the reason path from the current item to the next item. In the whole model, these reason paths will feed to the item attention module. Therefore, our model can recommend with learned sequential information.

5 RELATED WORK

Early recommendation systems mostly rely on Collaborative Filtering (CF) [25, 26], which are based on the idea that users with similar history will be more likely to purchase similar items. However, CF-based recommendations always suffer from sparsity issues

and cold-start problems. Therefore, some works utilize side information, like user and item attributes [6], item contents [21, 31] and other external information [39] to solve these issues. Among them, knowledge graph-based methods [32–34, 45] show great advantages to recommendation performance and explainability.

Knowledge-aware recommendations can be roughly categorized into embedding-based and path-based approaches. Prior efforts on embedding-based methods [42] always use embeddings of the knowledge graph to model the user-item interactions for recommendations. For example, exiting works [17] utilizes TranE [1] to embed user-item interactions to integrate knowledge into the recommendation system. Similarly, [7] embeds users and items into the same embedding space for recommendation. The above approaches model the relations among users and items using knowledge graph embedding methods, which achieved great improvements in model expressiveness. However, these methods are sensitive to the quality of related knowledge graphs.

To this end, meta-paths [27], and the connectivity among different types of nodes are introduced to recommendations, which have the ability to learn the explicit expression along each meta-path schema. In [43], it introduces Matrix Factorization (MF) and Factorization Machine (FM) to learn similarities generated by each meta-path for recommendation. [10] models rich objects and relations in knowledge graph and extracts different aspect-level similarity matrices thanks to meta-paths for the top-N recommendation. Although they achieved appealing performance, the limitation is still obvious that structured meta-path based similarity latent factors can only reflect mutual infectivity along meta-path schemas in a graph but cannot capture the specific information along particular path instances, which limits the explainability of recommendation.

More recently, injecting meta-paths as recommendation context (aggregation of meta-path instances) [16] was introduced for top-N recommendation. It provides fine-grained explanations based on specific instances. However, it ignores the important sequential dynamics of user-item interactions, limiting the performance of the recommendation performance and interpretability. To model sequentially, [32] utilizes LSTM to model the sequential information, but it only sequentially models path between users and items and ignores the importance of the user's clicked history sequences,

which are highly informative to infer user's preferences. To tackle the issue, [45] attempts to model the sequences of user's behaviours and path connectivity between users and items for recommendation. Nevertheless, it only models user-item paths, which ignores the item-item intrinsic relation information and cannot learn complex semantic information between items. Based on the above research, we propose a meta-path based recommendation with differentiated user-item and sequential item-item instances to enhance the learning ability for explainability recommendation.

6 CONCLUSION

We propose TMER, which explicitly models dynamic user-item interactions over time with path-based knowledge-aware explainable capabilities. We explore item-item paths between consecutive items with attention mechanisms for users' sequential behaviour modelling. Future works may include the following directions: 1) generate human-readable explanations for recommendation with NLP techniques, and 2) explore causality learning [36] to discover more appealing paths for explainablity.

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*. 2787–2795.

[2] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *WWW*. 1583–1592.

[3] Hongxu Chen, Hongzhi Yin, Tong Chen, Weiqing Wang, Xue Li, and Xia Hu. 2020. Social Boosted Recommendation with Folded Bipartite Network Embedding. *TKDE* (2020).

[4] Hongxu Chen, Hongzhi Yin, Weiqing Wang, Hao Wang, Quoc Viet Hung Nguyen, and Xue Li. 2018. PME: projected metric embedding on heterogeneous networks for link prediction. In *SIGKDD*. 1177–1186.

[5] Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. 2020. Try This Instead: Personalized and Interpretable Substitute Recommendation. *SIGIR* (2020).

[6] Songjie Gong. 2009. Employing User Attribute and Item Attribute to Enhance the Collaborative Filtering Recommendation. *JSW* 4, 8 (2009), 883–890.

[7] László Grad-Gyenge, Attila Kiss, and Peter Filzmoser. 2017. Graph embedding based recommendation techniques on the knowledge graph. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. 354–359.

[8] Pan Gu, Yuqiang Han, Wei Gao, Guandong Xu, and Jian Wu. 2020. Enhancing session-based social recommendation through item graph embedding and contextual friendship modeling. *Neurocomputing* 419 (2020), 190–202.

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NIPS*.

[10] Xiaotian Han, Chuan Shi, Senzhang Wang, S Yu Philip, and Li Song. 2018. Aspect-Level Deep Collaborative Filtering via Heterogeneous Information Networks.. In *IJCAI*. 3393–3399.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[12] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. 355–364.

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[14] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *CIKM*. 843–852.

[15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[16] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *SIGKDD*. 1531–1540.

[17] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR*. 505–514.

[18] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*. 135–142.

[19] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. In *ICDM*. IEEE, 1053–1058.

[20] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*. 931–940.

[21] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. 2002. Content-boosted collaborative filtering for improved recommendations. *AAAI* 23 (2002), 187–192.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[23] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP-IJCNLP*.

[24] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *SIGKDD*. 701–710.

[25] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.

[26] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. Springer, 291–324.

[27] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.

[28] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. 1067–1077.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv* (2017).

[31] Dongjing Wang, Shuiguang Deng, and Guandong Xu. 2018. Sequence-based context-aware music recommendation. *Information Retrieval Journal* 21, 2-3 (2018), 230–252.

[32] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*, Vol. 33. 5329–5336.

[33] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced Negative Sampling over Knowledge Graph for Recommendation. *arXiv preprint arXiv:2003.05753* (2020).

[34] Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *SIGIR*. 285–294.

[35] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S Sheng S. Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. 2019. Recurrent convolutional neural network for sequential recommendation. In *The Web Conference*.

[36] Guandong Xu, Tri Dung Duong, Qian Li, Shaowu Liu, and Xianzhi Wang. 2020. Causality Learning: A New Perspective for Interpretable Machine Learning. arXiv:2006.16789

[37] Guandong Xu, Yanhui Gu, Peter Dolog, Yanchun Zhang, and Masaru Kitsuregawa. 2011. Semrec: A semantic enhancement framework for tag based recommendation. In *AAAI*, Vol. 25.

[38] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, Jiali Yang, and Xiaofang Zhou. 2019. Social influence-based group representation learning for group recommendation. In *ICDE*. IEEE, 566–577.

[39] Hongzhi Yin, Weiqing Wang, Hao Wang, Ling Chen, and Xiaofang Zhou. 2017. Spatial-aware hierarchical collaborative deep learning for POI recommendation. *TKDE* 29, 11 (2017), 2537–2551.

[40] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *ICDM*. 283–292.

[41] Quan Yuan, Li Chen, and Shiwan Zhao. 2011. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *RecSys*. 245–252.

[42] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*.

[43] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *SIGKDD*. 635–644.

[44] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. 2020. Leveraging Demonstrations for Reinforcement Recommendation Reasoning over Knowledge Graphs. In *SIGIR*. 239–248.

[45] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2020. A Knowledge-Aware Attentional Reasoning Network for Recommendation.. In *AAAI*. 6999–7006.