Knowledge-Aware Explainable Reciprocal Recommendation

Kai-Huang Lai[1], Zhe-Rui Yang[1], Pei-Yuan Lai[2], Chang-Dong Wang[1*], Mohsen Guizani[3], Min Chen[4,5]

[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2]South China Technology Commercialization Center, Guangzhou, China
[3]Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE
[4]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[5]Pazhou Lab, Guangzhou, China
laikh6@mail2.sysu.edu.cn, yangzhr9@mail2.sysu.edu.cn, lpy@sctcc.cn, changdongwang@hotmail.com, mohsen.guizani@mbzuai.ac.ae, minchen@ieee.org

Abstract

Reciprocal recommender systems (RRS) have been widely used in online platforms such as online dating and recruitment. They can simultaneously fulfill the needs of both parties involved in the recommendation process. Due to the inherent nature of the task, interaction data is relatively sparse compared to other recommendation tasks. Existing works mainly address this issue through content-based recommendation methods. However, these methods often implicitly model textual information from a unified perspective, making it challenging to capture the distinct intentions held by each party, which further leads to limited performance and the lack of interpretability. In this paper, we propose a Knowledge-Aware Explainable Reciprocal Recommender System (KAERR), which models metapaths between two parties independently, considering their respective perspectives and requirements. Various metapaths are fused using an attention-based mechanism, where the attention weights unveil dual-perspective preferences and provide recommendation explanations for both parties. Extensive experiments on two real-world datasets from diverse scenarios demonstrate that the proposed model outperforms state-of-the-art baselines, while also delivering compelling reasons for recommendations to both parties.

Introduction

Reciprocal recommender systems (RRS) (Pizzato et al. 2010) have become increasingly popular in various online platforms such as online dating (Neve and Palomares 2019; Xia et al. 2016) and job recruitment (Jiang et al. 2020; Yang et al. 2022). Unlike traditional recommender systems that make uni-directional recommendations to users, RRS aims to fulfill the bilateral needs between two parties by making reciprocal recommendations (e.g. recommending satisfactory date partners to each other, matching job seekers and recruiters, etc.).

However, building an effective RRS faces unique challenges compared to traditional recommender systems. One major issue is the sparsity of interaction data. For example, in job recommendation, once a job seeker accepts an offer, the interaction between the job seeker and the recruiter becomes inactive until the job seeker looks for new jobs. Meanwhile, as the job position gets filled, the recruiter will stop interacting with candidates for that position. Such bidirectional inactivation after a successful matching leads to significantly fewer historical interaction signals for accurately modeling the preferences of both sides, compared to the abundant user-item interactions in traditional recommender systems.

To alleviate the data sparsity issue, existing works (Akehurst et al. 2011; Luo et al. 2020) have explored leveraging side information such as resumes and job postings. However, they rely on textual data with inconsistent formats from both sides. The free-form nature of such user-generated content makes it difficult to precisely extract and match preferences. In addition, they treat the information in a unified view without distinguishing between the two parties involved. However, the two sides often have distinct intentions and preferences when evaluating the same content. The inability to capture such dual perspectives from inconsistent data formats limits the accuracy and interpretability of existing models.

These limitations highlight the need for modeling dual perspectives in reciprocal recommendation. For instance, in job matching scenarios, a candidate and recruiter may align on certain dimensions like skills and industry, but have mismatches in other dimensions like location and education preferences due to their different focuses. Capturing such nuanced differences in intentions and motivations is crucial for improving the accuracy of matches.

To address the limitations of existing methods, in this paper, we propose a Knowledge-Aware Explainable Reciprocal Recommender System (KAERR) that incorporates side information from both parties involved in the recommendation process in a knowledge graph. By extracting metapaths between the two parties, KAERR can explicitly capture their distinct preferences and intentions. We encode the metapaths from dual perspectives using a bidirectional LSTM (Hochreiter and Schmidhuber 1996) and fuse them with an attention

*Corresponding author

mechanism to distill important signals differently. In addition, we make reciprocal predictions and optimize the model with a bilateral quadruple-based loss function. The learned attention weights also provide explainability by revealing the relative importance of different metapaths. By effectively modeling the knowledge graph information from dual perspectives, KAERR can improve recommendation accuracy and provide explanations. Extensive experiments verify the effectiveness of KAERR over state-of-the-art baselines.

The contributions of our work are summarized as follows:

- We propose a novel Knowledge-Aware Explainable Reciprocal Recommender System (KAERR) that models metapaths from a knowledge graph independently from the dual perspectives of the two parties involved using a bidirectional LSTM encoder.
- Extensive experiments on two real-world datasets demonstrate that KAERR consistently outperforms state-of-the-art baselines on reciprocal recommendation tasks.
- To the best of our knowledge, our model is the first reciprocal recommender system that can provide compelling reasons for the recommendations to both parties by revealing the relative importance of different metapaths through attention weights.

Related Work

Reciprocal Recommendation

Existing RRS studies can be grouped into several categories based on their methodology, including collaborative filtering-based methods (Cai et al. 2012; Xia et al. 2016; Neve and Palomares 2019), content-based methods (Alanazi and Bain 2013; Akehurst et al. 2011; Yang et al. 2017), hybrid methods (Zhou et al. 2023), and sequential-based methods (Zheng et al. 2023a). Collaborative filtering methods use past user interactions to infer preferences from similar patterns, but struggle with minimal history (cold-start). Content-based approaches need detailed text data to match user profiles, relying on the quality of this data. Hybrid methods combine behavior and content analysis to improve recommendations. Sequential methods use neural networks for sequence matching, yet also require extensive interaction history. Generally, current RRS inadequately address sparse interactions or the mutual aspect between two sides. Developing an approach that can overcome sparse bilateral signals and suit the reciprocal setting remains an open challenge.

Knowledge-Aware Recommendation

Knowledge graphs offer valuable context by mapping entities and their relationships, improving recommender systems' representation learning. Knowledge-aware recommendation techniques fall into three categories: embedding-based methods (Zhang et al. 2016; Wang et al. 2018; Cao et al. 2019) use entity and relation embeddings from knowledge graphs in user and item representations; path-based methods (Wu, Zhang, and Lin 2022; Li et al. 2022) extract knowledge graph metapaths to understand user-item connections; and GNN-based methods (Wu, Zhang, and Lin 2022; Li et al. 2022) utilize graph neural networks to learn from knowledge graph structures. While these methods enrich semantics, they are not specifically tailored for reciprocal recommendations and fail to differentiate the distinct intentions and preferences of both parties involved.

Explainable Recommendation

Explainable recommendation is a key research area with diverse explanation styles, such as predefined templates (Li, Chen, and Dong 2021), ranked sentences (Li, Zhang, and Chen 2021), knowledge graph paths (Xian et al. 2019), reasoning rules (Shi et al. 2020), and generated natural language (Li, Zhang, and Chen 2020). These styles range from using fixed templates and selected review sentences to leveraging knowledge graph semantics, inference rules, and language models to create tailored explanations. Nevertheless, most systems generate generic explanations without considering the unique needs of each party in a reciprocal recommendation scenario, a significant drawback for reciprocal recommendations where individual motivations and priorities vary greatly.

Preliminaries and Notations

To facilitate discussion in the following sections, we take the example of an online recruitment platform. Next, we formally define the notations for the concepts involved.

Bilateral Interaction Assume that we have a set of candidates $\mathcal{C} = \{c_1, c_2, \cdots, c_M\}$ and a set of jobs $\mathcal{J} = \{j_1, j_2, \cdots, j_N\}$ posted by recruiters, where M and N are the total numbers of candidates and jobs. Each candidate or recruiter can send requests to jobs or candidates that meet their criteria. All accepted requests form a matching set $\mathcal{M} = \{(c_i, j_k) \mid c_i \in \mathcal{C}, j_k \in \mathcal{J}\}$. Rejected requests lead to unilateral matches, which are recorded in matrix $\mathbf{U}^{M\times N}$, where $u_{ik} = 1$ means candidate c_i applied for job j_k but got rejected, and $u_{ik} = -1$ means recruiter of job j_k invited candidate c_i but got declined, and the default value within the matrix is 0.

Knowledge Graph We construct a knowledge graph $\mathcal{G} = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ represents the set of entities including candidates, jobs and their attributes, and $\mathcal{R}$ represents the set of relations between entities. Each triplet (h, r, t) denotes a head entity h, relation r and tail entity t. For an example, (CandidateA, HasSkill, Java) indicates CandidateA has the skill Java.

Metapath A metapath is a sequence of entity types and relation types that defines a specific semantic path between entities. For instance, a metapath of "Candidate-HasSkill-Skill-RequireSkill-Job" reveals the skills that candidates possess which are required for certain jobs. We pre-define a set of metapath patterns and extract all metapath instances from the knowledge graph. A metapath $p \in \mathcal{P}$ can be denoted as $(e_1, r_1, e_2, \cdots, r_{n-1}, e_n)$, where $e_i \in \mathcal{E}$ represents entities and $r_j \in \mathcal{R}$ represents relations.

Problem Definition Given the bilateral interaction history $\mathcal{M}$, $\mathbf{U}$ and knowledge graph $\mathcal{G}$, our goal is to learn a matching function $f(c_i, j_k)$ that predicts the matching probability between candidate c_i and job j_k based on their interaction records and the metapaths $\mathcal{P}_{i,k}$ that connect them within the knowledge graph.

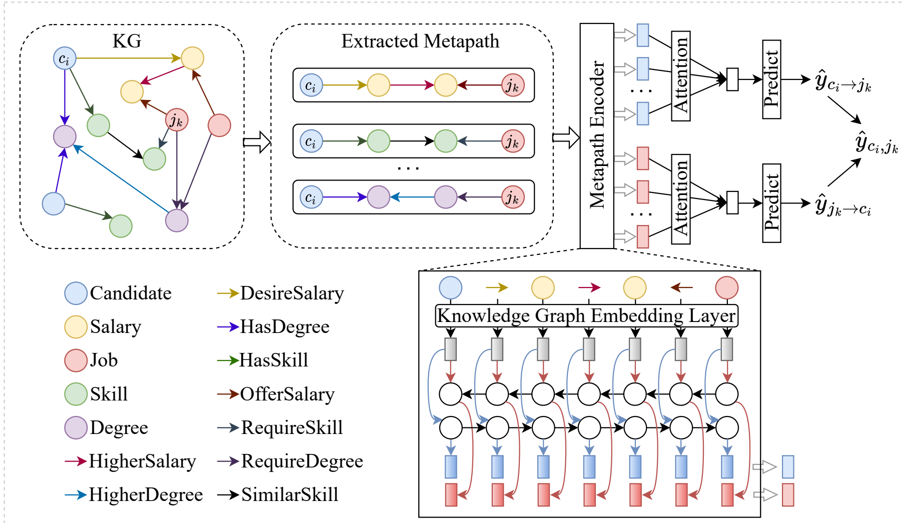


Figure 1: The overall framework of KAERR.

Method

In this section, we present the proposed **K**nowledge-**A**ware **E**xplainable **R**eciprocal **R**ecommender System (KAERR) (shown in Figure 1), which incorporates two main modules: 1) *Dual-Perspective Metapath Encoder*, which encodes the metapaths from the perspectives of both candidates and jobs independently using a BiLSTM encoder; 2) *Attentive Metapath Fusion*, which learns attention weights for each metapath and fuses the dual representations based on the attention weights. In addition to the above two modules, we adopt MLP-based methods to predict matching probabilities from both perspectives and average them as the final prediction. Finally, we optimize the model by minimizing a proposed bilateral quadruple-based loss that considers both bilateral matches and unilateral matches to enhance performance in reciprocal recommendation.

Dual-Perspective Metapath Encoder

To capture the distinct preferences of candidates and jobs over each metapath, we first encode the metapath instances from each side independently. This is because the same metapath may imply different intentions from two sides. For example, the metapath "Candidate-HasDegree-PhD-LowerDegree-Bachelor-RequireDegree-Job" indicates a positive signal for the recruiter that the candidate's education level meets the requirement. However, it may not be that important or even negative for the candidate who pursues a higher degree. Therefore, modeling metapaths from dual perspectives is necessary.

We choose to use a BiLSTM encoder because each metapath can be seen as a sequence consisting of entities and relations. LSTM is adept at feature extraction from sequential data, including the ability to handle dependencies within sequences. Our choice to opt for LSTM over Transformers (Vaswani et al. 2017) is motivated by the need for computational efficiency and to mitigate overfitting risks, considerations that become significant in the context of processing metapaths that are typically short and exhibit limited variability. By treating the candidate and job as the start of the sequence respectively in two LSTM directions, the BiLSTM encoder is able to learn the dual-perspective representations for each metapath.

For modeling each metapath instance $p = (e_1, r_1, ..., r_{n-1}, e_n) \in \mathcal{P}_{i,k}$ between candidate c_i and job j_k, we first map the elements in the metapath to low-dimensional embeddings through a Knowledge Graph Embedding Layer initialized by TransR (Lin et al. 2015). TransR is able to capture the structural features of entities and relations, which facilitates subsequent metapath modeling.

Specifically, the embedding of the metapath instance is:

$$\mathbf{E} = Embed(p) = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_T], \tag{1}$$

where $\mathbf{E} \in \mathbf{R}^{T \times d_e}$, $\mathbf{e}_t \in \mathbf{R}^{d_e}$ is the d_e-dimensional knowledge graph embedding of the t-th element, and $T = 2n - 1$ is the length of metapath.

Then the embedding sequence $\mathbf{E}$ is fed into a bidirectional LSTM encoder to learn contextual representations:

$$\overrightarrow{\mathbf{h}}_t = LSTM(\mathbf{e}_t, \overrightarrow{\mathbf{h}}_{t-1}), \tag{2}$$

$$\overleftarrow{\mathbf{h}}_t = LSTM(\mathbf{e}_t, \overleftarrow{\mathbf{h}}_{t+1}), \tag{3}$$

where $\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t \in \mathbf{R}^{d_h}$ are the d_h-dimensional forward and backward hidden states at step t, respectively.

To represent the perspectives of candidate c_i and job j_k over the metapath instance p, we compute dual-perspective aggregated representations by averaging the BiLSTM hidden states from two directions:

$$\mathbf{p}^c = \frac{1}{T}\sum_{t=1}^{T} \overrightarrow{\mathbf{h}}_t, \quad \mathbf{p}^j = \frac{1}{T}\sum_{t=1}^{T} \overleftarrow{\mathbf{h}}_t, \tag{4}$$

where $\mathbf{p}^c, \mathbf{p}^j \in \mathbf{R}^{d_h}$ are the metapath representations of candidate c_i and job j_k respectively.

By encoding the metapath sequentially from two ends, the BiLSTM model is able to learn distinct preferences over the same metapath from dual perspectives.

Attentive Metapath Fusion

For the same metapath instance, the attention assigned by the candidate and recruiter sides may differ, as it implies distinct intentions for them. To capture such dual-perspective preferences, we adopt an attention mechanism for metapath representation aggregation. The attention module can learn soft weights to highlight influential metapaths while suppressing irrelevant ones differently for the two sides.

Specifically, for each candidate-job pair (c_i, j_k), given their metapath representations $\{\mathbf{p}_l^c\}_{l=1}^L$ and $\{\mathbf{p}_l^j\}_{l=1}^L$ from the dual-perspective metapath encoder, where L is the number of metapaths, we compute the attention weights as:

$$\alpha_l^c = \sigma(\mathbf{p}_l^c \mathbf{w}_c^a + b_c^a), \quad \alpha_l^j = \sigma(\mathbf{p}_l^j \mathbf{w}_j^a + b_j^a), \tag{5}$$

where $\mathbf{w}_c^a, \mathbf{w}_j^a \in \mathbf{R}^{d_h}$ and $b_c^a, b_j^a \in \mathbf{R}$ are trainable weight vectors and bias terms, and $\sigma(\cdot)$ is the sigmoid function that squashes the attention weights between 0 and 1 for soft selection:

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{6}$$

The fused metapath representations are computed as weighted sums using the attention weights:

$$\mathbf{m}^c = \sum_{l=1}^{L} \alpha_l^c \mathbf{p}_l^c, \quad \mathbf{m}^j = \sum_{l=1}^{L} \alpha_l^j \mathbf{p}_l^j, \tag{7}$$

where $\mathbf{m}^c$ and $\mathbf{m}^j$ represent the aggregated candidate and job representations that imply their preferences over each other.

The attention weights α_l^c and α_l^j indicate the relative importance of different metapaths from the dual perspectives, which provides explanations for the recommendation results.

Prediction

Unlike the traditional recommender systems which make predictions by fusing representations from both sides, we make dual-perspective predictions separately and then average them.

Specifically, given the aggregated metapath representations $\mathbf{m}^c$ and $\mathbf{m}^j$ of candidate c_i and job j_k, we have:

$$\hat{y}_{c_i \to j_k} = \sigma(\mathbf{m}^c \mathbf{w}_c^p + b_c^p), \quad \hat{y}_{j_k \to c_i} = \sigma(\mathbf{m}^j \mathbf{w}_j^p + b_j^p), \tag{8}$$

where $\mathbf{w}_c^p, \mathbf{w}_j^p \in \mathbf{R}^d$ are trainable weight vectors that transform aggregated metapath representations into matching probabilities. $b_c^p, b_j^p \in \mathbf{R}$ are trainable bias terms. And $\hat{y}_{c_i \to j_k}$ predicts the probability of job j_k satisfying candidate c_i based on the candidate's aggregated preferences over metapaths, while $\hat{y}_{j_k \to c_i}$ predicts the probability in the opposite direction.

To combine the dual-perspective predictions, we take their average as the final matching probability:

$$\hat{y}_{i,k} = \frac{1}{2}(\hat{y}_{c_i \to j_k} + \hat{y}_{j_k \to c_i}). \tag{9}$$

Optimization

To optimize the model parameters, we propose a bilateral quadruple loss that incorporates bilateral matching loss and unilateral matching loss.

Follow the previous work (Yang et al. 2022), for each positive sample match $\langle c_i, j_k \rangle$, we construct negative samples $\langle c_i, j_k' \rangle$ and $\langle c_i', j_k \rangle$, where c_i' and j_k' are randomly sampled negative candidate and job respectively. The training set can be denoted as $\mathcal{D} = \{(i,k,i',k') \mid (i,k) \in \mathcal{M}, (i,k') \in \overline{\mathcal{M}}, (i',k) \in \overline{\mathcal{M}}\}$, where $\mathcal{M}$ and $\overline{\mathcal{M}}$ are the matched and unmatched sets, and (i,k,i',k') is the abbreviation of quadruple (c_i, j_k, c_i', j_k').

The bilateral matching loss is defined as:

$$\mathcal{L}_{bm} = -\frac{1}{|\mathcal{D}|} \sum_{(i,k,i',k') \in \mathcal{D}} \log\left(\sigma\left(2\hat{y}_{i,k} - \hat{y}_{i,k'} - \hat{y}_{i',k}\right)\right), \tag{10}$$

where σ denotes the sigmoid function.

The negative samples may contain some unilaterally matched ones, which can be identified by the unilateral match matrix $\mathbf{U}$. Specifically, $u_{ik} = 1$ indicates candidate c_i applied but got rejected by job j_k, $u_{ik} = -1$ indicates the opposite direction, and $u_{ik} = 0$ means no unilateral match between them. The unilateral matching loss is defined as:

$$\mathcal{L}_{um} = -\frac{1}{|\mathcal{D}|} \sum_{(i,k,i',k') \in \mathcal{D}} \log(\sigma(f(i,k,i',k'))), \tag{11}$$

where

$$f(i,k,i',k') = u_{ik'}(\hat{y}_{i \to k'} - \hat{y}_{k' \to i}) + u_{i'k}(\hat{y}_{i' \to k} - \hat{y}_{k \to i'}). \tag{12}$$

By combining the two parts, the final loss function is:

$$\mathcal{L} = \mathcal{L}_{bm} + \lambda \mathcal{L}_{um}, \tag{13}$$

where λ balances the two loss terms. By minimizing this loss function, the model parameters can be optimized to improve the matching prediction performance.

Compared to the previous methods that use either cross-entropy loss or pairwise loss, our bilateral quadruple-based loss models the reciprocal matching in two directions simultaneously, and also accommodates unilateral matches in reciprocal recommendation.

Experienments

To answer the following questions, we conduct experiments on two real-world datasets from different scenarios. Our code is available at: https://github.com/AllminerLab/Code-for-KAERR-master.

- **RQ1:** How does our model perform compared to the existing state-of-the-art methods?
- **RQ2:** What are the contributions of different components of our model to the overall performance?
- **RQ3:** How do parameters influence the results of KAERR?
- **RQ4:** Can our model provide intuitive explanations for the prediction results?

Dataset	Zhaopin	UEM
# Candidates/Researchers	4,500	3,124
# Jobs/Demands	19,114	6,247
# Interactions	29,792	18,960
Sparsity	99.97%	99.90%
# Match	28,195	11,245
# KG Entity Types	9	11
# KG Relation Types	24	13
# KG Entities	35,471	58,214
# KG Relations	431,831	585,794

Table 1: Statistics of the experimental datasets.

Datasets We evaluate our model on two real-world datasets from different reciprocal recommendation scenarios. The overall statistics are shown in Table 1.

- **Online Recruitment**. We use a dataset from the Aliyun Programming Competition on Person-Job Fitting[1], provided by a large Chinese online recruitment platform, namely Zhaopin. For simplicity, the dataset is called Zhaopin. In this dataset, if a job seeker views and applies for a job posting, and the recruiter accepts the application, this candidate-job pair is treated as a positive match, indicating mutual satisfaction.
- **University-Enterprise Matching**. We have compiled a dataset derived from industry collaboration records spanning the last five years at Sun Yat-sen University. For simplicity, the dataset is called UEM. In this scenario, we need to recommend suitable university researchers for the technology demands proposed by enterprises. Meanwhile, we also need to recommend appropriate enterprise demands for the researchers based on their capabilities.

Baselines We conduct experiments to compare our proposed model with the following baseline methods:

- **BPRMF** (Rendle et al. 2012) is a matrix factorization model that learns user and item representations by optimizing a pairwise Bayesian Personalized Ranking loss.
- **NCF** (He et al. 2017) replaces the inner product in matrix factorization with a multi-layer perceptron, which helps to capture non-linear relationships.
- **LFRR** (Neve and Palomares 2019) is a latent factor model adapted for reciprocal recommendation.
- **LightGCN** (He et al. 2020) is a simplified graph convolutional network for recommendation that captures collaborative filtering signals to generate personalized recommendations efficiently.
- **PJFNN** (Zhu et al. 2018) is a convolutional neural network model for person-job fit prediction. It learns joint representations of person and job from historical application data in an end-to-end manner.
- **BPJFNN** (Qin et al. 2018)is an RNN-based model for person-job fit prediction. It uses BiLSTM to derive semantic representations for job requirements and applicant experiences.
- **APJFNN** (Qin et al. 2018) employs hierarchical attention on RNN-derived job and applicant representations to identify key requirements and relevant experiences.
- **DPGNN** (Zhou et al. 2023) uses graph representation learning with two nodes per entity to capture two-way selection preferences and interactions.

We categorize the baseline models into three groups according to their core techniques: (1) Collaborative filtering methods including BPRMF, NCF, LFRR and LightGCN, which make recommendations based on user-item interactions; (2) Content-based methods including PJFNN, BPJFNN, and APJFNN, which rely on profile content features; (3) Hybrid method DPGNN that combines collaborative filtering and content-based filtering. Except for BPRMF and NCF, all the other baseline models are proposed specifically for the reciprocal recommendation scenario. It's crucial to highlight that we omitted comparisons with sequential recommendation models like ReSeq (Zheng et al. 2023b) due to their dependence on extensive interaction histories and sequential data, requirements that our dataset does not meet.

Evaluation Following (Yang et al. 2022), we adopt four common ranking metrics: Recall (R@k), Precision (P@k), Normalized Discounted Cumulative Gain (NDCG@k) and Mean Reciprocal Rank (MRR@k). We set k to 5 for evaluation. We perform evaluation from both sides simultaneously for each positive match, which is well suited for reciprocal recommendation. Specifically, for each positive match, we sample 20 negative instances for both sides to construct two ranking lists. We then report the average ranking metrics across both lists.

Implementation Details We implement the baseline models using RecBole (Zhao et al. 2022) library. Hyper-parameters for all methods are tuned through grid search. The Adam optimizer is utilized for model training. The learning rate is selected from $\{0.01, 0.001, 0.0001\}$ via tuning. Early stopping with a patience of 10 epochs is adopted to prevent overfitting.

Performance Comparison (RQ1)

Table 2 presents the comparison results. It can be observed that collaborative filtering-based baselines perform poorly due to limitations in modeling sparse interactions. Although content-based baselines achieve some improvements compared to collaborative filtering methods, they still underperform the hybrid methods. The hybrid method DPGNN achieves the second-best performance across all metrics, indicating that utilizing both text descriptions and interactions is important.

In comparison, our proposed KAERR method achieves superior performance over all baselines on both datasets. Unlike the existing methods, KAERR explicitly models the metapaths from dual perspectives and fuses them with attention weights, which allows for capturing the distinct intentions of each party and focusing on influential metapaths. This leads to more accurate matching between the two sides.

[1]https://tianchi.aliyun.com/dataset/31623

Dataset	Perspective	Candidates/Researchers				Jobs/Demands			
	Metric	R@5	P@5	NDCG@5	MRR@5	R@5	P@5	NDCG@5	MRR@5
Zhaopin	BPRMF	0.2769	0.0570	0.2164	0.1997	0.3500	0.0789	0.2576	0.2367
	NCF	0.3606	0.0739	0.2378	0.2012	0.3236	0.0731	0.2257	0.2020
	LFRR	0.2833	0.0582	0.2215	0.2045	0.3545	0.0802	0.2577	0.2352
	LightGCN	0.2981	0.0611	0.2312	0.2089	0.3601	0.0814	0.2631	0.2393
	PJFNN	0.6929	0.1425	0.4984	0.4392	0.6468	0.1384	0.4605	0.4057
	BPJFNN	0.3056	0.0625	0.1970	0.1632	0.2318	0.0480	0.1389	0.1107
	APJFNN	0.3074	0.0631	0.1900	0.1536	0.2319	0.0485	0.1396	0.1116
	DPGNN	0.7777	0.1617	0.6144	0.5658	0.7460	0.1628	0.5869	0.5441
	KAERR	**0.9477**	**0.1979**	**0.8275**	**0.7895**	**0.9499**	**0.2059**	**0.8333**	**0.7990**
UEM	BPRMF	0.3202	0.0723	0.1987	0.2156	0.3825	0.0845	0.2663	0.2538
	NCF	0.3542	0.0721	0.2335	0.1998	0.3278	0.0756	0.2198	0.2047
	LFRR	0.3389	0.0698	0.2354	0.2496	0.3875	0.0925	0.2734	0.2547
	LightGCN	0.3458	0.0714	0.2402	0.2547	0.3928	0.0944	0.2792	0.2601
	PJFNN	0.7321	0.1623	0.5123	0.4578	0.6723	0.1502	0.4789	0.4156
	BPJFNN	0.7514	0.1687	0.5236	0.4629	0.6915	0.1552	0.4887	0.4486
	APJFNN	0.7587	0.1695	0.5287	0.4655	0.6963	0.1618	0.4894	0.4523
	DPGNN	0.8259	0.1897	0.6532	0.5923	0.8064	0.1921	0.6243	0.5940
	KAERR	**0.9146**	**0.1932**	**0.8014**	**0.7507**	**0.9202**	**0.1961**	**0.8128**	**0.7667**

Table 2: Performance comparison of all methods.

Dataset	Perspective	Candidates/Researchers				Jobs/Demands			
	Metric	R@5	P@5	NDCG@5	MRR@5	R@5	P@5	NDCG@5	MRR@5
Zhaopin	KAERR	**0.9477**	**0.1979**	**0.8275**	**0.7895**	**0.9499**	**0.2059**	**0.8333**	**0.7990**
	w/o DPME	0.8936	0.1531	0.7897	0.7612	0.9178	0.1736	0.7912	0.7714
	w/o AMF	0.9105	0.1582	0.7875	0.7624	0.9235	0.1821	0.7890	0.7727
	w/o BQL	0.9079	0.1597	0.7759	0.7595	0.9201	0.1799	0.7823	0.7612
UEM	KAERR	**0.9146**	**0.1732**	**0.8014**	**0.7507**	**0.9202**	**0.1961**	**0.8128**	**0.7667**
	w/o DPME	0.8653	0.1427	0.7652	0.7301	0.8827	0.1725	0.7698	0.7221
	w/o AMF	0.8702	0.1495	0.7781	0.7399	0.8785	0.1925	0.7754	0.7344
	w/o BQL	0.8669	0.1380	0.7664	0.7368	0.8802	0.1786	0.7721	0.7302

Table 3: Performance comparison between KAERR and its variants.

Ablation Study (RQ2)

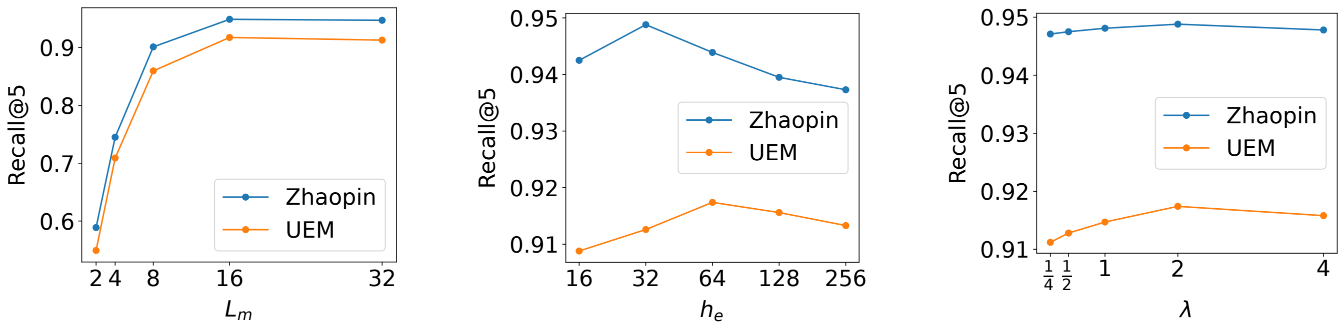
To verify the effectiveness of our proposed components, we conducted ablation studies by removing each of the key designs in KAERR. Specifically, we consider the following three variants of KAERR: (1) KAERR w/o DPME replaces the dual-perspective metapath encoder with a shared LSTM encoder, where candidates and jobs use a common metapath representation; (2) KAERR w/o AMF substituting the attentive metapath fusion by simple mean pooling; (3) KAERR w/o BQL changes the bilateral quadruple loss to the conventional BPR loss. The results in Table 3 demonstrate a performance decline when removing any of the above components. This confirms that all the components in KAERR make pivotal contributions to improving KAERR's performance.

Hyper-Parameter Analysis (RQ3)

The parameter tuning results are shown in Figure 2. We study the impacts of three key hyper-parameters: maximum number of metapaths L_m, knowledge graph embedding size h_e, and λ in the loss function. L_m controls the amount of semantics captured from the knowledge graph. Through testing $L_m \in 2, 4, 8, 16, 32$, we find that $L_m = 16$ achieves the best performance, as too small values lead to insufficient semantics while too large values incorporate noisy information. For h_e, the optimal value differs between the two datasets due to the knowledge graph size. λ balances the weights of bilateral and unilateral matching losses, and the best value is 2 since the proportion of unilateral matches is relatively small in the interaction data, and moderately increasing the coefficient that can helps to better exploit their information. This analysis provides insights into how to properly set these key factors.

Case Study (RQ4)

Figure 3 shows successful and unsuccessful Candidate-Job matches. In the successful match, our model predicts high scores from both the candidate's and the recruiter's per-



(a) L_m (b) h_e (c) λ

Figure 2: The performance of KAERR with different settings of L_m, h_e, and λ.

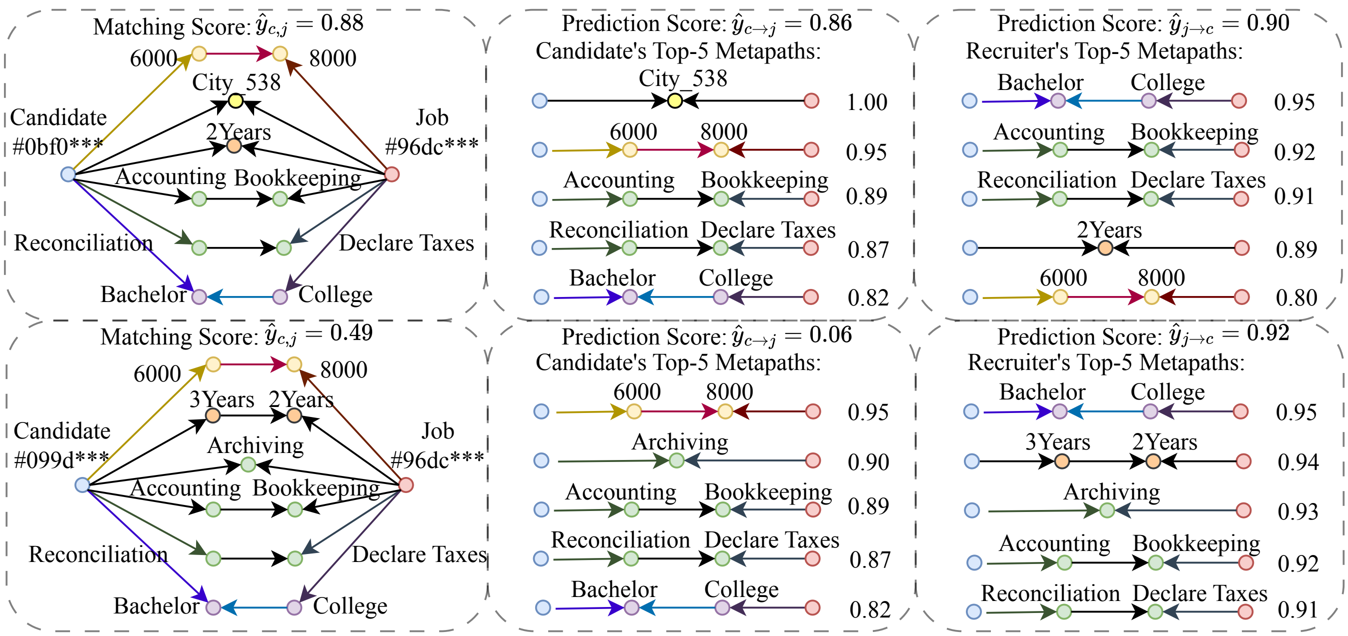


Figure 3: Examples of a successful Candidate-Job match (Top) and an unsuccessful Candidate-Job match (Bottom).

spectives. The attention weights on metapaths highlight the key factors influencing the match. For the candidate, the top weights are on location and salary, suggesting these are the primary considerations. On the other hand, the recruiter places more emphasis on education, skills, and experience.

In contrast, the unsuccessful match depicted at the bottom shows a different scenario. Even though the candidate satisfies the job requirements with suitable education, work experience, and skills, which is indicated by a high prediction score from the job's perspective, the candidate's own prediction score for the job remains low. This discrepancy in scores is due to the job's location not meeting the candidate's preferences, ultimately resulting in a low overall matching score. This example underscores the importance of considering both parties' preferences in the matching process and demonstrates the nuanced interpretability our model provides in real-world recommendation scenarios.

Conclusion

In this paper, we proposed a novel Knowledge-Aware Explainable Reciprocal Recommender System (KAERR) that effectively incorporates knowledge graph information to address the sparsity issue in the reciprocal recommendation. By extracting metapaths and modeling them from the dual perspectives of the two involved parties, KAERR is able to capture their distinct intentions and preferences. An attention mechanism is adopted to fuse the metapath representations by learning soft weights indicating the importance of each metapath. Extensive experiments on two real-world datasets verified that KAERR achieves state-of-the-art performance. Furthermore, the attention weights provide interpretability by revealing the relative influence of different metapaths. For future work, we plan to explore incorporating metapath modeling with other graph learning techniques to capture more information from knowledge graphs.

Acknowledgements

This work was supported by Guangdong Basic and Applied Basic Research Foundation (2022B1515120059), NSFC (62276277 and 62276109), and Guangdong Provincial Engineering Research Center of Intelligent Matching for Technology Commercialization (2022A175). And Mohsen Guizani appreciates the research support provided by the Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI) (8481000021).

References

Akehurst, J.; Koprinska, I.; Yacef, K.; Pizzato, L. A. S.; Kay, J.; and Rej, T. 2011. CCR - A Content-Collaborative Reciprocal Recommender for Online Dating. In *IJCAI*, 2199–2204.

Alanazi, A.; and Bain, M. 2013. A people-to-people content-based reciprocal recommender using hidden markov models. In *RecSys*, 303–306.

Cai, X.; Bain, M.; Krzywicki, A.; Wobcke, W.; Kim, Y. S.; Compton, P.; and Mahidadia, A. 2012. Reciprocal and Heterogeneous Link Prediction in Social Networks. In *PAKDD (2)*, 193–204.

Cao, Y.; Wang, X.; He, X.; Hu, Z.; and Chua, T. 2019. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences. In *WWW*, 151–161.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*, 639–648.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T. 2017. Neural Collaborative Filtering. In *WWW*, 173–182.

Hochreiter, S.; and Schmidhuber, J. 1996. LSTM can Solve Hard Long Time Lag Problems. In *NIPS*, 473–479.

Jiang, J.; Ye, S.; Wang, W.; Xu, J.; and Luo, X. 2020. Learning Effective Representations for Person-Job Fit by Feature Fusion. In *CIKM*, 2549–2556.

Li, L.; Chen, L.; and Dong, R. 2021. CAESAR: context-aware explanation based on supervised attention for service recommendations. *J. Intell. Inf. Syst.*, 57(1): 147–170.

Li, L.; Zhang, Y.; and Chen, L. 2020. Generate Neural Template Explanations for Recommendation. In *CIKM*, 755–764.

Li, L.; Zhang, Y.; and Chen, L. 2021. EXTRA: Explanation Ranking Datasets for Explainable Recommendation. In *SIGIR*, 2463–2469.

Li, T.; Li, X.; Wang, C.; Li, X.; Gao, S.; and Han, D. 2022. FF-KGAT: Feature Fusion Based Knowledge Graph Attention Network for Recommendation. In *AIPR*, 468–474.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*, 2181–2187.

Luo, L.; Yang, L.; Xin, J.; Fang, Y.; Zhang, X.; Yang, X.; Chen, K.; Zhang, Z.; and Liu, K. 2020. RRCN: A Reinforced Random Convolutional Network based Reciprocal Recommendation Approach for Online Dating. *CoRR*, abs/2011.12586.

Neve, J.; and Palomares, I. 2019. Latent factor models and aggregation operators for collaborative filtering in reciprocal recommender systems. In *RecSys*, 219–227.

Pizzato, L. A. S.; Rej, T.; Chung, T.; Koprinska, I.; and Kay, J. 2010. RECON: a reciprocal recommender for online dating. In *RecSys*, 207–214.

Qin, C.; Zhu, H.; Xu, T.; Zhu, C.; Jiang, L.; Chen, E.; and Xiong, H. 2018. Enhancing Person-Job Fit for Talent Recruitment: An Ability-aware Neural Network Approach. In *SIGIR*, 25–34.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *CoRR*, abs/1205.2618.

Shi, S.; Chen, H.; Ma, W.; Mao, J.; Zhang, M.; and Zhang, Y. 2020. Neural Logic Reasoning. In *CIKM*, 1365–1374.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NIPS*, 30.

Wang, H.; Zhang, F.; Xie, X.; and Guo, M. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *WWW*, 1835–1844.

Wu, Z.; Zhang, X.; and Lin, X. 2022. KGAT: Predicting Drug-Target Interaction Based on Knowledge Graph Attention Network. In *ICIC (2)*, 438–450.

Xia, P.; Zhai, S.; Liu, B.; Sun, Y.; and Chen, C. X. 2016. Design of reciprocal recommendation systems for online dating. *Soc. Netw. Anal. Min.*, 6(1): 32:1–32:16.

Xian, Y.; Fu, Z.; Muthukrishnan, S.; de Melo, G.; and Zhang, Y. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. In *SIGIR*, 285–294.

Yang, C.; Hou, Y.; Song, Y.; Zhang, T.; Wen, J.; and Zhao, W. X. 2022. Modeling Two-Way Selection Preference for Person-Job Fit. In *RecSys*, 102–112.

Yang, S.; Korayem, M.; AlJadda, K.; Grainger, T.; and Natarajan, S. 2017. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. *Knowl. Based Syst.*, 136: 37–45.

Zhang, F.; Yuan, N. J.; Lian, D.; Xie, X.; and Ma, W. 2016. Collaborative Knowledge Base Embedding for Recommender Systems. In *KDD*, 353–362.

Zhao, W. X.; Hou, Y.; Pan, X.; Yang, C.; Zhang, Z.; Lin, Z.; Zhang, J.; Bian, S.; Tang, J.; Sun, W.; Chen, Y.; Xu, L.; Zhang, G.; Tian, Z.; Tian, C.; Mu, S.; Fan, X.; Chen, X.; and Wen, J. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In *CIKM*, 4722–4726.

Zheng, B.; Hou, Y.; Zhao, W. X.; Song, Y.; and Zhu, H. 2023a. Reciprocal Sequential Recommendation. *CoRR*, abs/2306.14712.

Zheng, B.; Hou, Y.; Zhao, W. X.; Song, Y.; and Zhu, H. 2023b. Reciprocal Sequential Recommendation. In *RecSys*, 89–100.

Zhou, L.; Chen, W.; Zeng, D.; Cheng, S.; Liu, W.; Zhang, M.; and Qu, H. 2023. DPGNN: Dual-perception graph neural network for representation learning. *Knowl. Based Syst.*, 268: 110377.

Zhu, C.; Zhu, H.; Xiong, H.; Ma, C.; Xie, F.; Ding, P.; and Li, P. 2018. Person-Job Fit: Adapting the Right Talent for the Right Job with Joint Representation Learning. *ACM Trans. Manag. Inf. Syst.*, 9(3): 12:1–12:17.