KGAT: Knowledge Graph Attention Network for Recommendation

Xiang Wang[†], Xiangnan He[‡], Yixin Cao[†], Meng Liu[*], Tat-Seng Chua[†]

[†] National University of Singapore, [‡] University of Science and Technology of China, [*] Shandong University

Collaborative Knowledge Graph (CKG)

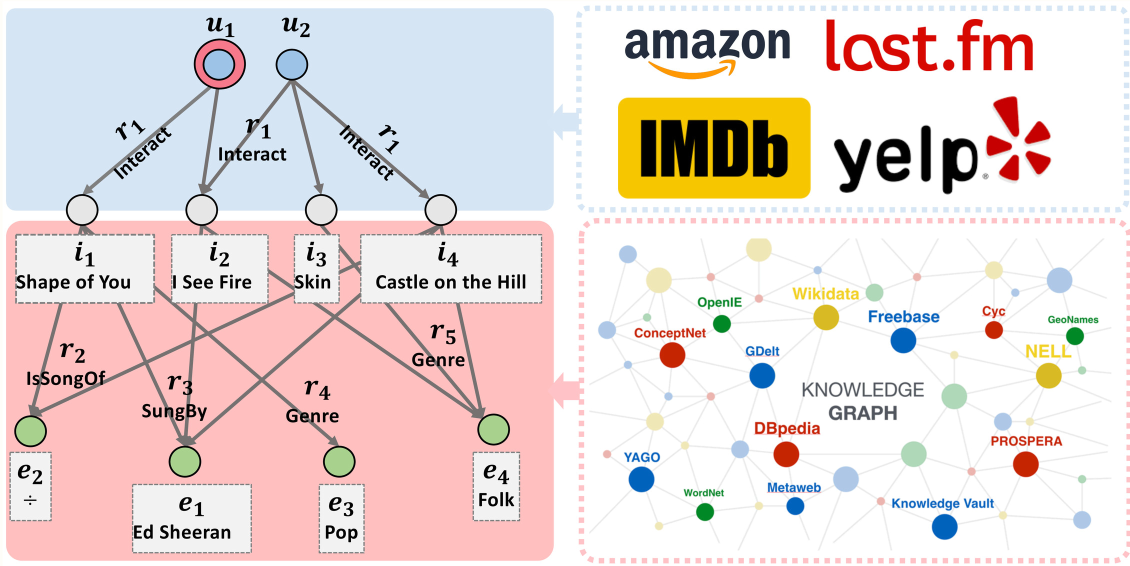


Figure 1: Incorporating knowledge graph into user-item bipartite graph.

Components of CKG

User-Item Bipartite Graph

- User-Item Direct Interactions

$u_1 \xrightarrow{r_1} i_1$

+

Knowledge Graph

- Item-Item External Connections

$i_1 \xrightarrow{r_2} e_1$

=

Collaborative Knowledge Graph

- High-order connectivity between users and items

$u_1 \xrightarrow{r_1} i_1 \xrightarrow{r_2} e_1 \xrightarrow{-r_2} i_2 \Rightarrow u_1 \xrightarrow{r_1} i_2$

- Possible reasons on potential recommendations

Figure 2: Interlinks of CKG, especially high-order relations, bring benefits to recommendation and explanations.

How Prior Works Leverage CKG for Recommendation?

Summary & Limitations of Three-type Works

	Supervised Learning-based	Path-based	Regularization-based
Knowledge Usage	Item knowledge → a generic feature vector	Connectivity → paths connecting users & items	Graph structure → an additional item representations or loss
Relation Usage	-	To define meta-path Or select qualified paths	To regularize the learning of KG embeddings
Limitations	• Fail to capture CF signals • Ignore semantic & structure information	• Require labor-intensive feature engineering • Have rather high complexity	• Lack explicit modeling of high-order relations
Examples	FM, NFM, TEM, Wide&Deep ...	MCRec, RippleNet, FMG, KPRN ...	KTUP, CFKG, CKE ...

Figure 3: Due to the characteristics of these models, high-order relations have not been fully and properly explored.

Our Goal: Develop a model that can exploit high-order information in KG in an efficient, explicit, and end-to-end manner.

How to Achieve High-order Relation Modeling in an Explicit & End-to-End Manner?

Inspired by the recent success of graph neural networks, we propose Knowledge Graph Attention Network (KGAT) for KG-based Recommendation.

Knowledge Graph Attention Network (KGAT)

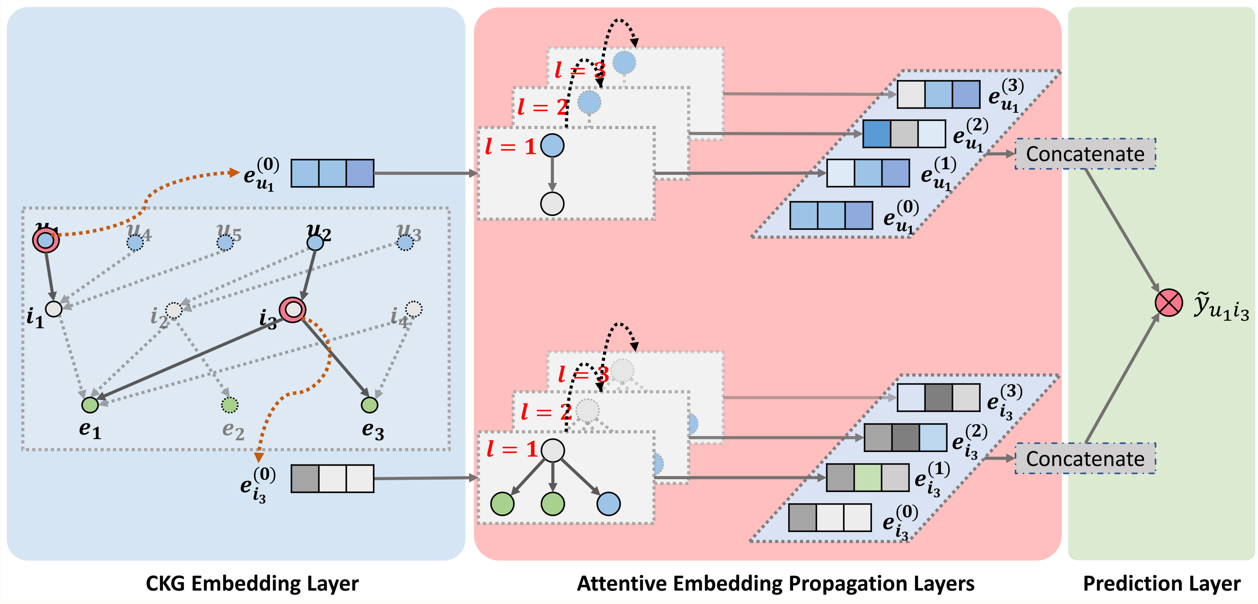


Figure 4: Illustration of the proposed KGAT model.

I. CKG Embedding Layer

We adopt TransR to parameterize entities and relations of CKG as vector representations, considering direct connectivity of each triplet (h, r, t).

$$g(h,r,t) = \|\mathbf{W}_r\mathbf{e}_h + \mathbf{e}_r - \mathbf{W}_r\mathbf{e}_t\|_2^2, \tag{1}$$

$$\mathcal{L}_{\text{KG}} = \sum_{(h,r,t,t')\in\mathcal{T}} -\ln\sigma\Big(g(h,r,t') - g(h,r,t)\Big). \tag{2}$$

II. Attention Embedding Propagation Layer

1. **Information Propagation:** We perform information propagation between an entity h and its neighbors $\mathcal{N}_h$:

$$\mathbf{e}_{\mathcal{N}_h} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)\mathbf{e}_t, \tag{3}$$

where $\pi(h,r,t)$ controls how much information being propagated from tail entity t to head entity h conditioned to relation r.

2. **Knowledge-aware Attention:** We implement $\pi(h,r,t)$ via relational attention mechanism, which is formulated as follows:

$$\pi(h,r,t) = (\mathbf{W}_r\mathbf{e}_t)^\top \tanh\Big((\mathbf{W}_r\mathbf{e}_h + \mathbf{e}_r)\Big). \tag{4}$$

3. **Information Aggregation:** The final phase is to aggregate the entity representation $\mathbf{e}_h$ and its ego-network representations $\mathbf{e}_{\mathcal{N}_h}$ as the new representation of entity h:

$$\mathbf{e}_h^{(1)} = \text{LeakyReLU}\Big(\mathbf{W}_1(\mathbf{e}_h + \mathbf{e}_{\mathcal{N}_h})\Big) + \text{LeakyReLU}\Big(\mathbf{W}_2(\mathbf{e}_h \odot \mathbf{e}_{\mathcal{N}_h})\Big). \tag{5}$$

4. **High-order Propagation:** We can further stack more propagation layers to explore the high-order connectivity information, gathering the information propagated from the higher-hop neighbors:

$$\mathbf{e}_{\mathcal{N}_h}^{(l-1)} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)\mathbf{e}_t^{(l-1)}. \tag{6}$$

III. Model Prediction

After performing L layers, we obtain multiple representations for user node u, namely $\{\mathbf{e}_u^{(1)}, \cdots, \mathbf{e}_u^{(L)}\}$; analogous to item node i,

$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)}\|\cdots\|\mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)}\|\cdots\|\mathbf{e}_i^{(L)}. \tag{7}$$

Finally, we conduct inner product of user and item representations, so as to predict their matching score:

$$\hat{y}(u,i) = \mathbf{e}_u^{*\top}\mathbf{e}_i^*, \tag{8}$$

$$\mathcal{L}_{\text{CF}} = \sum_{(u,i,j)\in\mathcal{O}} -\ln\sigma\Big(\hat{y}(u,i) - \hat{y}(u,j)\Big). \tag{9}$$

IV. Model Optimization

Finally, we have the objective function to learn Equations (2) and (9) jointly, as follows:

$$\mathcal{L}_{\text{KGAT}} = \mathcal{L}_{\text{KG}} + \mathcal{L}_{\text{CF}}. \tag{10}$$

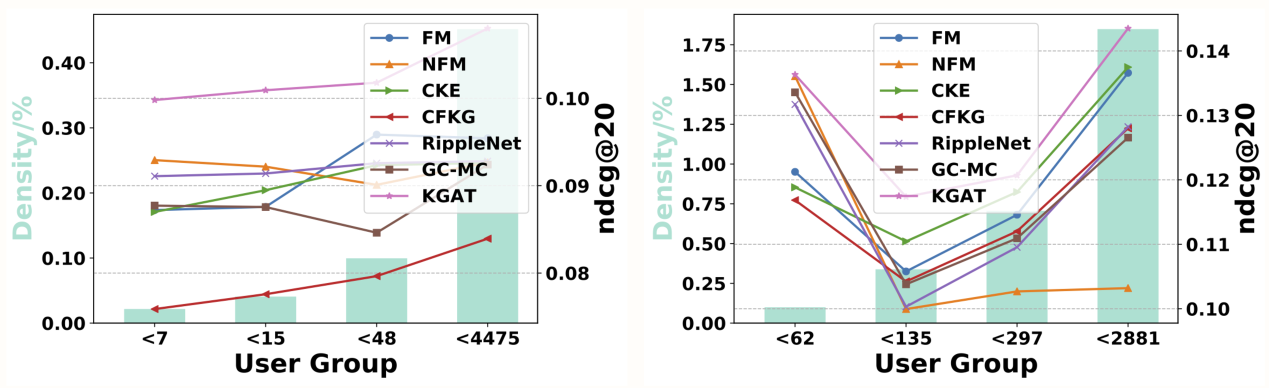
Results

Overall Performance Comparison

	Amazon-Book		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
FM	0.1345	0.0886	0.0778	0.1181	0.0627	0.0768
NFM	**0.1366**	0.0913	**0.0829**	0.1214	0.0660	0.0810
CKE	0.1343	0.0885	0.0736	0.1184	0.0657	0.0805
CFKG	0.1142	0.0770	0.0723	0.1143	0.0522	0.0644
MCRec	0.1113	0.0783	-	-	-	-
RippleNet	0.1336	0.0910	0.0791	0.1238	**0.0664**	**0.0822**
GC-MC	0.1316	0.0874	0.0818	**0.1253**	0.0659	0.0790
KGAT	**0.1489***	**0.1006***	**0.0870***	**0.1325***	**0.0712***	**0.0867***
%Improv.	8.95%	10.05%	4.93%	5.77%	7.18%	5.54%

Figure 5: KGAT consistently yields the best performance on all the datasets.

Interaction Sparsity Levels



(a) ndcg on Amazon-Book (b) ndcg on Last-FM

Figure 6: KGAT outperforms the other models in most cases, especially on the two sparsest user groups in Amazon-Book and Yelp2018.

Case Study for Explainable Recommendation

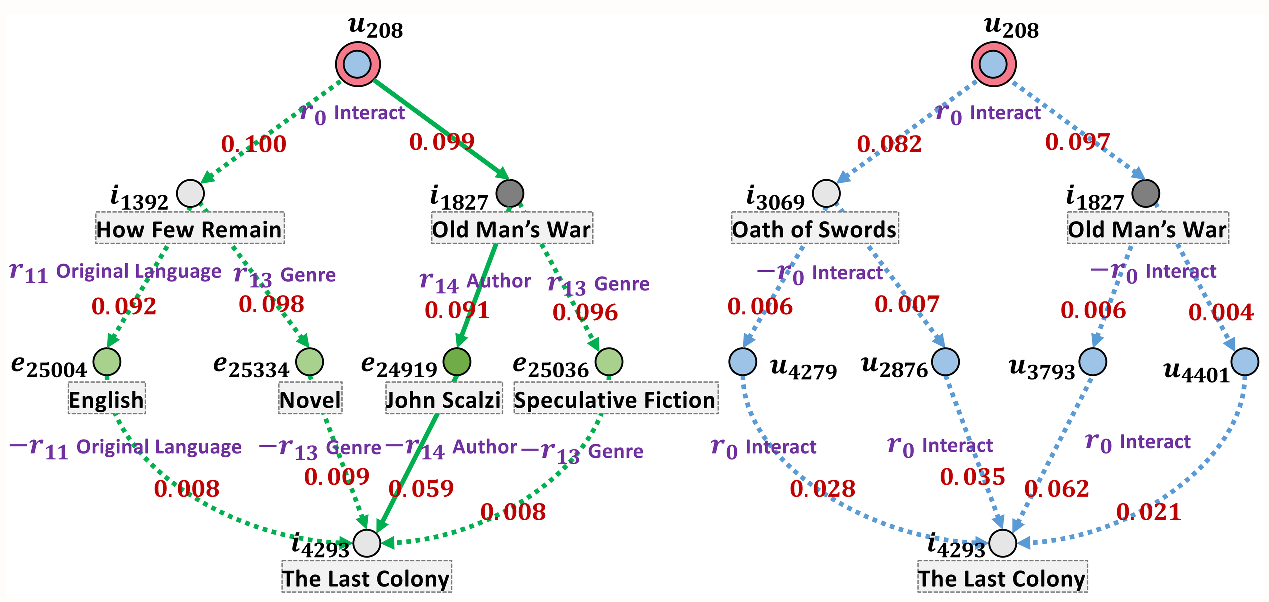


Figure 7: KGAT captures the behavior-based and attribute-based high-order connectivity, which play a key role to infer user preferences.

Datasets & Codes

Figure 8: Scan me to get three public datasets and KGAT codes.