Paper Review

KGAT: Knowledge Graph Attention Network

M1 Yang Boming

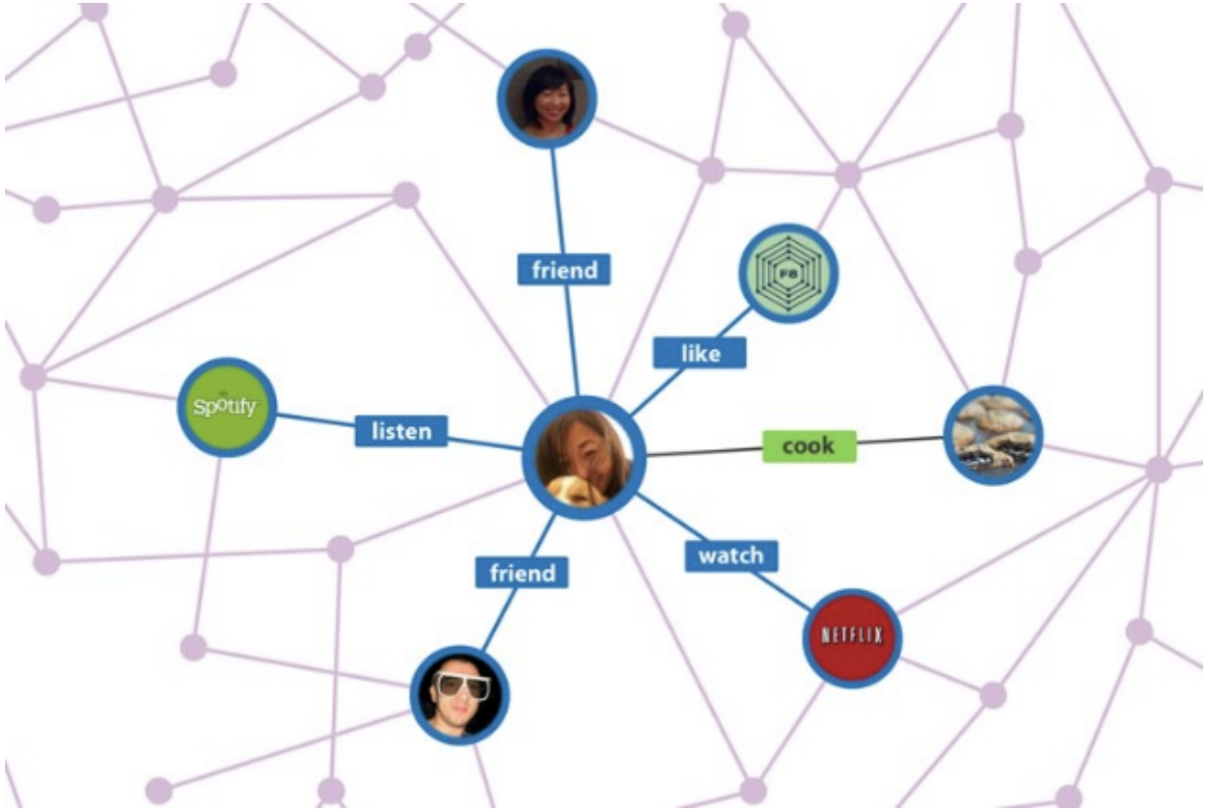
Background

Knwoledge Graph

1. Heterogeneous graph: node or two or more types of edges.
2. node becomes an **entity** and edge becomes a **relationship**.
3. Traditional Graph $G = (V, E)$

   Knowledge Graph $G = (V, E, R, T)$

   Nodes, Edges, Node type, Relation type.



https://velog.io/@jbeen2/CS224W-17.-Reasoning-over-Knowledge-Graphs

Abstract

Knowledge Graph Attention Network(KGAT)

1. Provide better recommendation with item side information.
2. KGAT model, which achieves high-order relation modeling in an explicit and end-top-end manner under GNN.
3. Conduct extensive experiments to demonstrate the effectiveness pf KGAT and ites interpretability

Task Formulation

Collaborative Knowledge Graph (CKG)

- CKG: which encodes user behaviors and item knowledge as a unified relational graph.

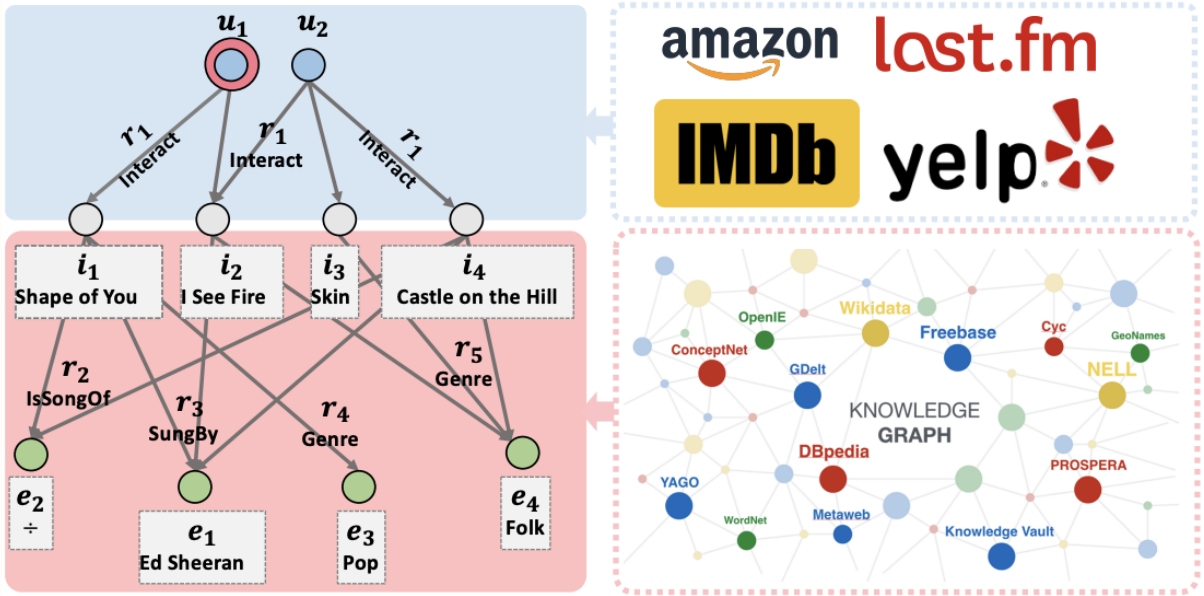


Figure 1: Incorporating knowledge graph into user-item bipartite graph.

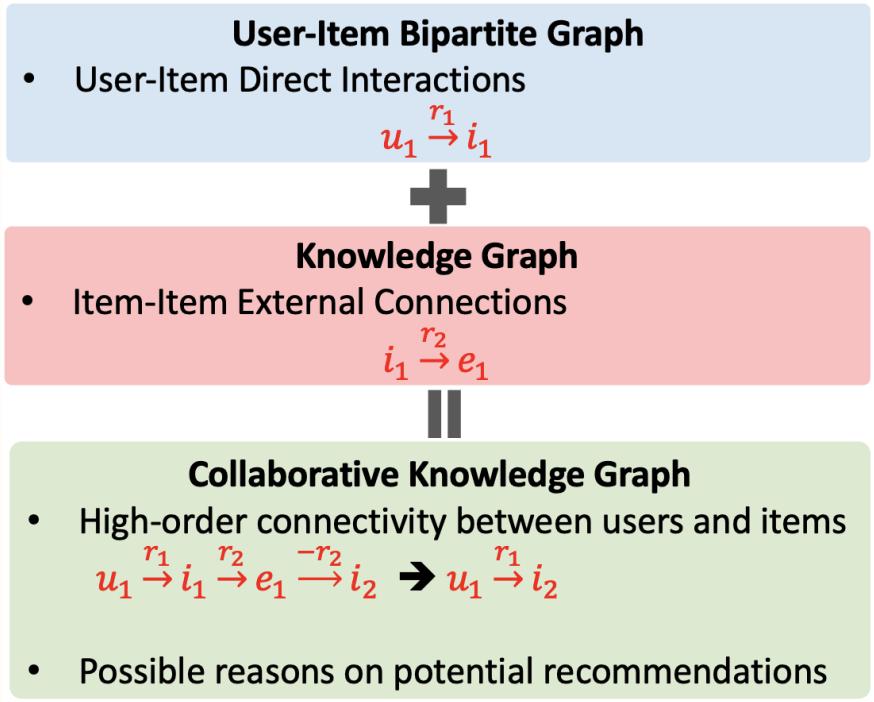


Figure 2: Interlinks of CKG, especially high-order relations, bring benefits to recommendation and explanations.

http://staff.ustc.edu.cn/~hexn/slides/kdd19-KGAT-poster.pdf

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.

http://staff.ustc.edu.cn/~hexn/slides/kdd19-KGAT-poster.pdf

Methodology

Overview of KGAT

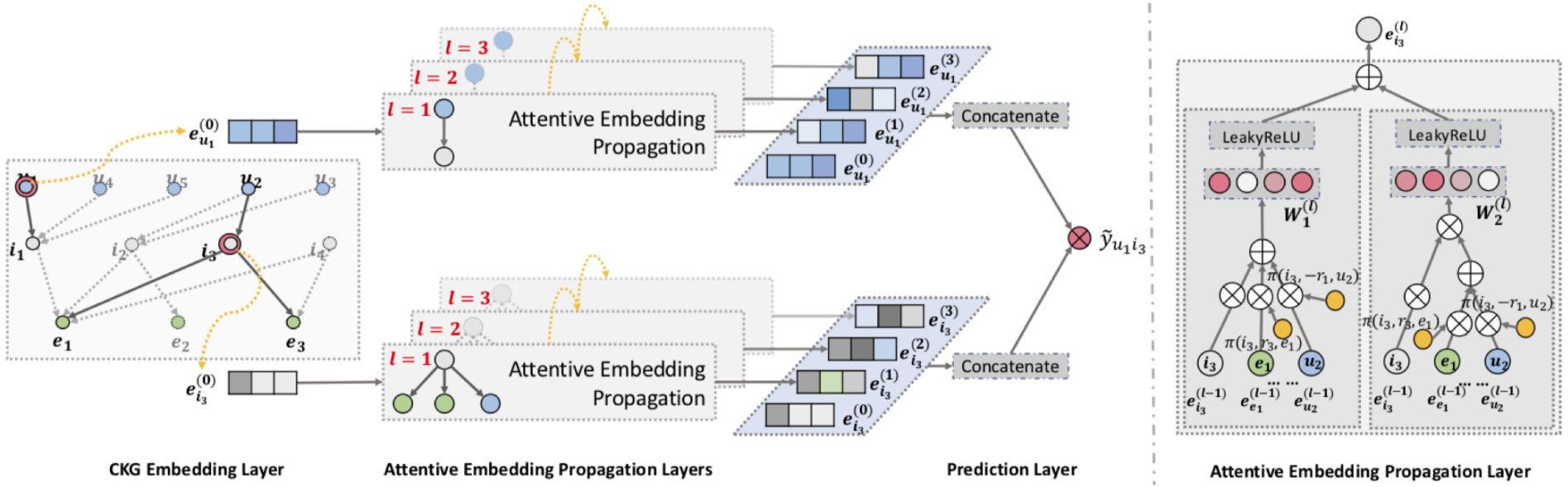


Figure 2: Illustration of the proposed KGAT model. The left subfigure shows model framework of KGAT, and the right subfigure presents the attentive embedding propagation layer of KGAT.

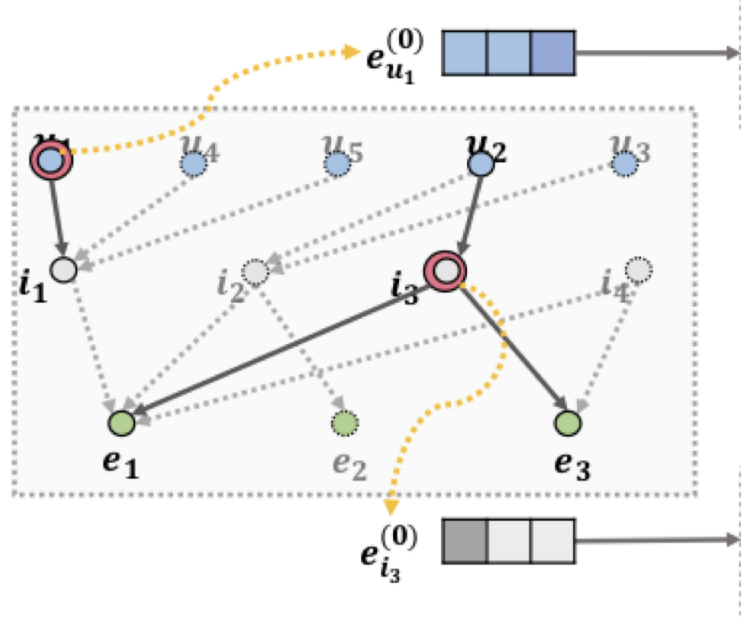
Methodology

CKG Embedding Layer

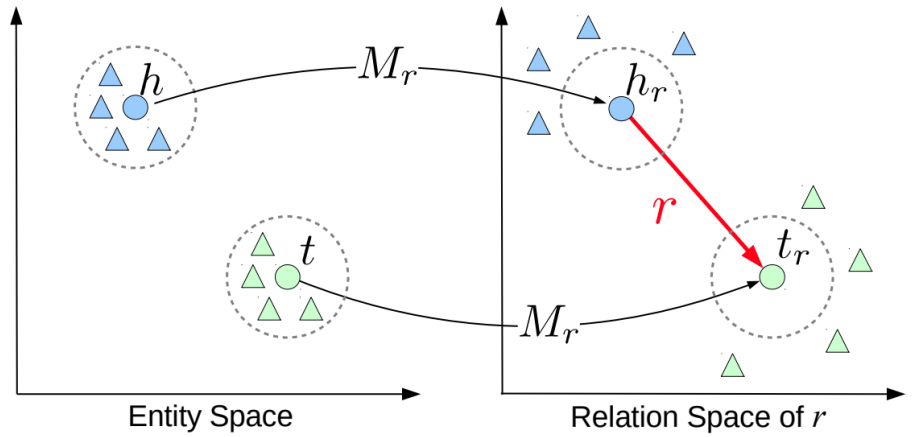
We adopy TransR to parameterize entities and relations of CKG as vector represetations, considering direct connectivity of each triple (h, r, t):

$$g(h, r, t) = \| W_r e_h + e_r - W_r e_t \|_2^2$$

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



CKG Embedding Layer



Attentive Embedding Propagation Layer 1/2

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

   $e_{\mathcal{N}_h} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)e_t$

   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**: Implement $\pi(h,r,t)$ via relational attention mechanism, which is formulated as follows:

   $\pi(h,r,t) = (W_r e_t)\top tanh((W_r e_h + e_r))$

Attentive Embedding Propagation Layer 2/2

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

$$e_h^{(1)} = LeakyReLU(W_1(e_h + e_{\mathcal{N}_h})) + LeakyReLU(W_2(e_h \odot e_{\mathcal{N}_h}))$$

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

$$e_{\mathcal{N}_h}^{(l-1)} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)e_t^{(l-1)}$$

Model Prediction

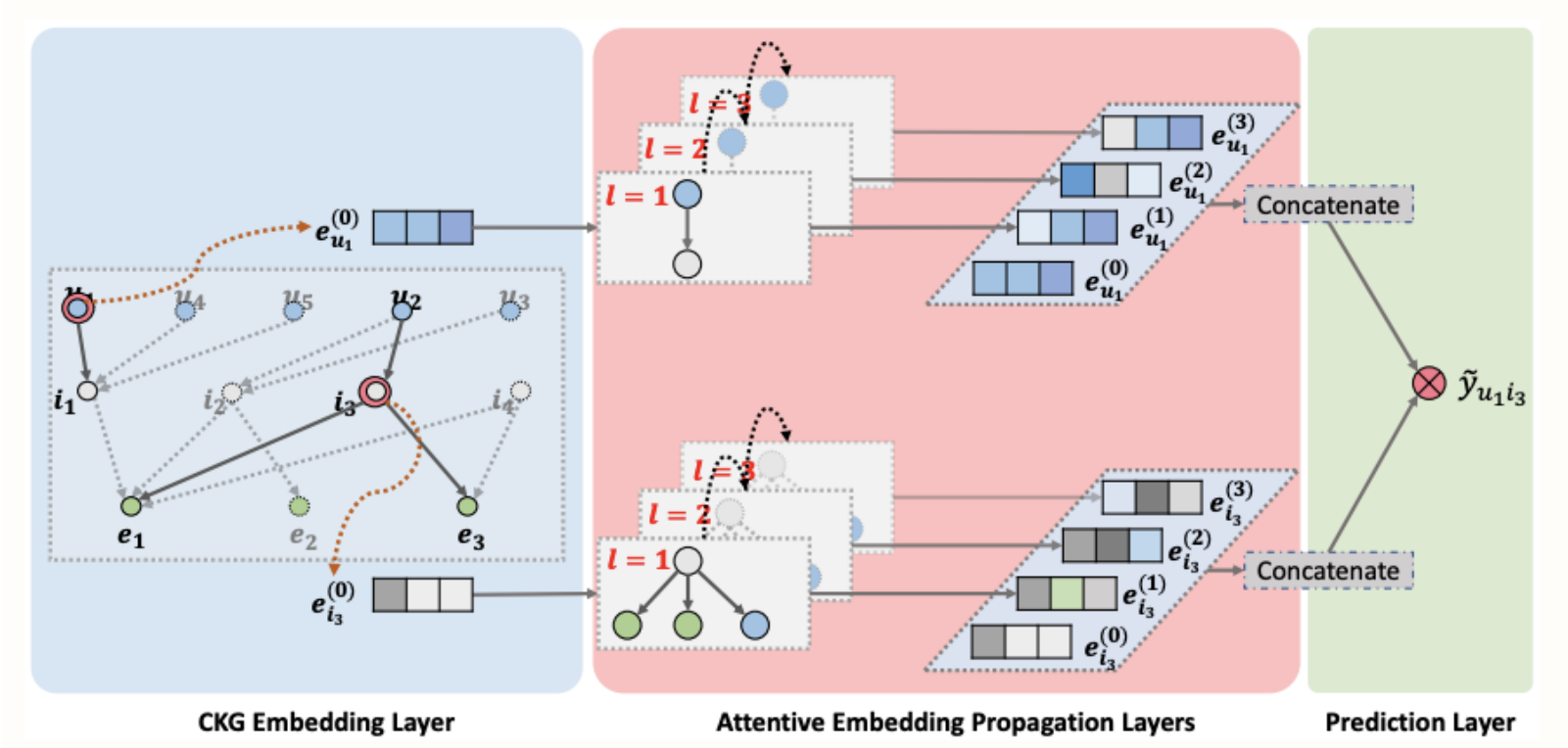
1. After performing L layers, obtain multiple representations for user node u, namely $e_u^{(1)}, \cdots, e_u^{(L)}$:
   $e_u^* = e_u^{(0)} \parallel \cdots \parallel e_u^{(L)}, e_i^* = e_i^{(0)} \parallel \cdots \parallel e_i^{(L)}$
2. Finally, conduct inner product of user and item representations, so as to predict their matching score:

$$\hat{y}(u,i) = e_u^* \top e_i^*$$
$$\mathcal{L}_{CF} = \sum_{(u,i,j)\in\mathcal{O}} -\ln \sigma(\hat{y}(u,i) - \hat{y}(u,j))$$

Optimization

Finally, the objective function to learn the loss function of KG and the loss function of CF joinlty, as follows:

$$\mathcal{L}_{KGAT} = \mathcal{L}_{KG} + \mathcal{L}_{CF}$$



Result

Overall Performance Comparsion

- KGAT consistently yields the best performance on all the datasets.

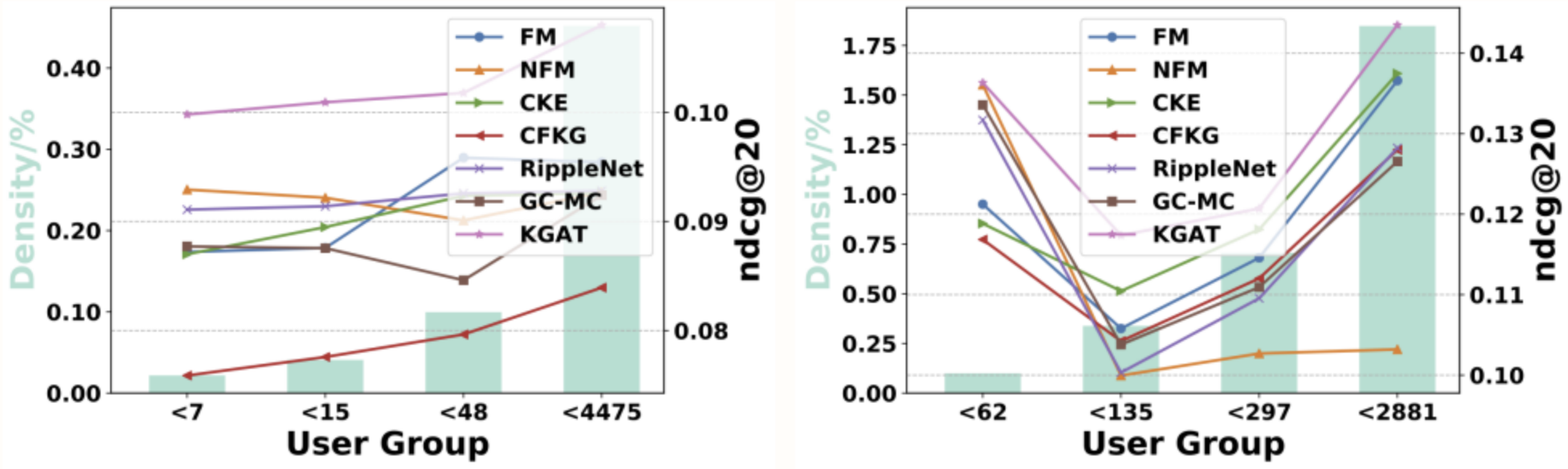
	Amazon-Book		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
FM	0.1345	0.0886	0.0778	0.1181	0.0627	0.0768
NFM	**0.1366**	0.0913	**0.0829**	0.1214	0.0660	0.0810
CKE	0.1343	0.0885	0.0736	0.1184	0.0657	0.0805
CFKG	0.1142	0.0770	0.0723	0.1143	0.0522	0.0644
MCRec	0.1113	0.0783	-	-	-	-
RippleNet	0.1336	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.

Result

Interaction Sparsity Levels

- Sparsity issue usually limits the expressiveness of recommender systems.
  (The number on x axis mean the interaction numbers per user)



(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.

Result

Case Study for Explainable Recommendation

- Solid line in left subfigure from u_{208} to i_{4293}, has the highest attention score.

  The explannation as *The Last Colony is recommended since you have watched Old Man's War written by the same author John Scalzi*

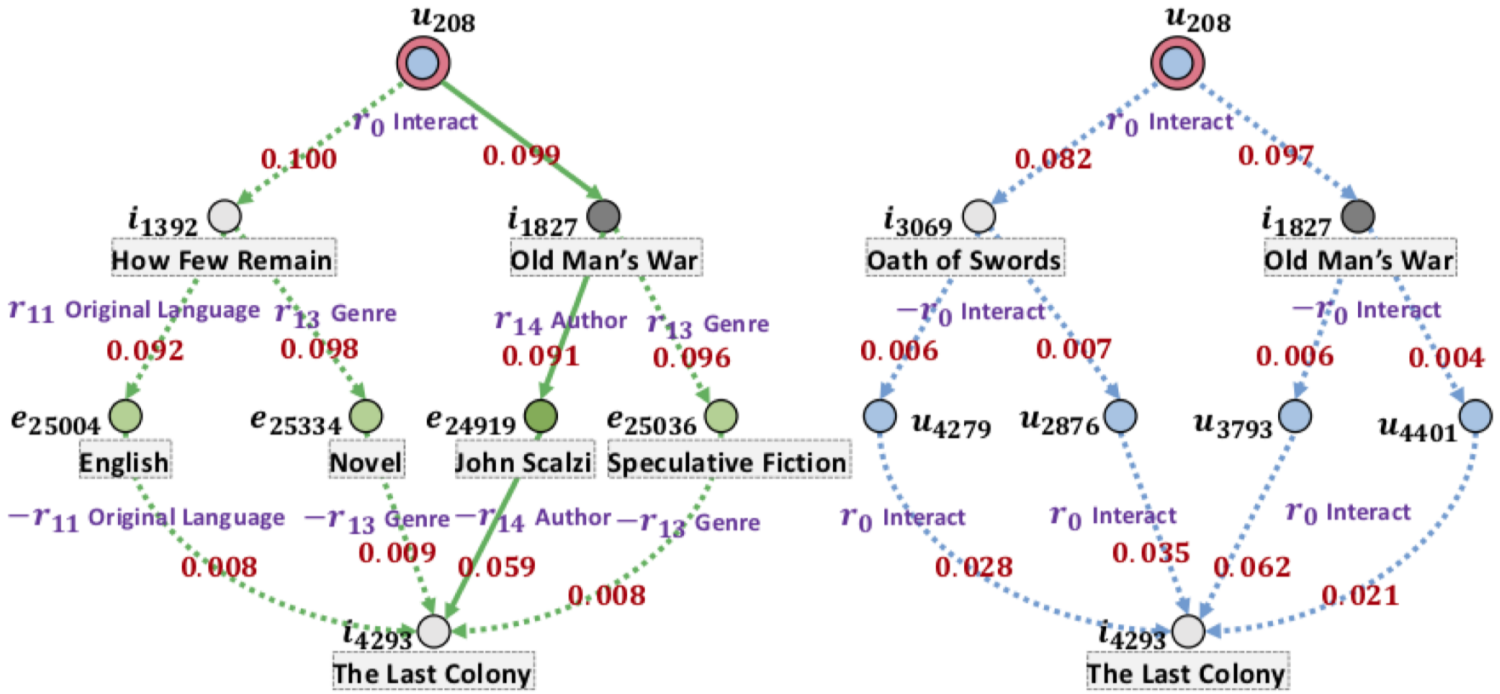


Figure 4: Real Example from Amazon-Book.

Conclusion

- High-order connectivity with semantic relations in CKG for knowledge-aware recommendation.
- Core: the attentive embedding propagation layer, which adaptively propagates the embeddings from a node's neighbors to update the node's representation.
- Extensive experiments on three real-world datasets demonstrate the rationality and effectiveness of KGAT.

Coding Task

Pull Request

- Implements other models
- Trains other datasets
- Paper review documents update
- Dockerfile updates
- Bugs and fixes