Enhanced Knowledge Graph Embedding for Multi-Task Recommendation via Integrating Attribute Information and High-Order Connectivity

Yani Wang
wangyani@nuaa.edu.cn
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu, China

Aoran Li
2469705253@qq.com
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu, China

Ji Zhang
Ji.Zhang@usq.edu.au
University of Southern Queensland
Toowoomba, Queensland, Australia

Bohan Li*
bhli@nuaa.edu.cn
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu, China

ABSTRACT

Recently, knowledge graph (KG) has been introduced into recommender systems as side information to mitigate problems of sparsity and cold start, which attracts growing attention. Among these works, the multi-task learning methods that learn KG-related tasks and recommendation tasks have greatly enhanced the effectiveness of the recommendation. However, existing works tend to ignore attribute triples in the real knowledge graph, which makes the research affected by the sparsity of KGs. General knowledge graph embedding methods in multi-task recommendation barely take direct relations between entities into account, that leads to the ignorance of rich information in high-order relations. In addition, the lack of consideration on the user attribute information makes recommendation less than satisfactory. In order to relieve these issues, we propose a model of enhanced knowledge graph embedding for recommendation based on the multi-task learning, HMKR. Our method alternately trains recommendation task considering user-side information and enhanced knowledge graph embedding task integrating attribute information and high-order connectivity. The experiments on the real dataset MovieLens verify the effectiveness of HMKR. Even in the sparse data scenarios, the performance of our model is also satisfactory.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender Systems; Knowledge Graph; Multi-Task Learning; Graph Neural Network

*Corresponding author

IJCKG'21, December 6–8, 2021, Virtual Event, Thailand

ACM ISBN 978-1-4503-9565-6/21/12…$15.00
https://doi.org/10.1145/3502223.3502242

ACM Reference Format:
Yani Wang, Aoran Li, Ji Zhang, and Bohan Li. 2021. Enhanced Knowledge Graph Embedding for Multi-Task Recommendation via Integrating Attribute Information and High-Order Connectivity. In *The 10th International Joint Conference on Knowledge Graphs (IJCKG'21), December 6–8, 2021, Virtual Event, Thailand*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3502223.3502242

1 INTRODUCTION

Currently, recommender systems are widely utilized to provide service for users through interacting with them. The recommender systems based on collaborative filtering have been extensively used [14]. They model user preferences through the resemblance of users or interactive items [17]. However, recommender systems based on collaborative filtering face sparsity and cold start problems [36]. To overcome the challenges, researchers have explored various types of side information. Recently, knowledge graph is introduced into the recommender systems as side information to relieve sparsity and cold start, which has caused growing attention.

Knowledge graph is a heterogeneous graph composed of entities, relations and attributes. The entity is the object of the real-world and the conceptual abstraction, relations represent the relationships between entities or the attributes of entities [11]. Attributes are inherent characteristics of entities. The KGs are usually encoded in the form of triples. The triples have two types [18], one is relation triples, e.g.,(*Li Bai*, *Friendof*, *Du Fu*), another is attribute triples, e.g.,(*Li Bai*, *Born*, *"701 A.D."*). The existing KG-based recommender systems adopt KG through three ways: embedding-based method, path-based method, and unified method [8]. Among them, the multi-task learning methods can learn KG-related tasks and recommendation tasks jointly so that knowledge graph can effectively enhance recommendation. Multi-task learning is a valuable learning method. It uses mutually significative information included in related tasks to help increase performance [35]. Papers [24], [2], [28] apply multi-task learning strategies to train recommendation and KG-related tasks jointly and find that joint learning improves performance and recommendation quality compared with individual learning.

However, the existing works tend to ignore massive attribute triples in the real knowledge graph [18], which makes the research affected by the sparsity of the KG. Knowledge graph embedding is a key foundation for various applications. The general knowledge graph embedding methods in multi-task recommendation barely take the direct relations between entities into consideration, but ignore the high-order structural relations, that will lead to the ignorance of rich information [15]. Meanwhile, user-side information is very significant [33], existing works lack consideration of user attribute information such as age and gender, and make recommendation less than satisfactory, especially when the user has very little interaction.

To relieve these issues, we propose a model of enhanced knowledge graph embedding for recommendation based on the multi-task learning [24], HMKR. Our method shares features between items and entities, and alternately trains recommendation and enhanced knowledge graph embedding. We consider the attribute triples which encode rich semantic information, and employ the idea of embedding propagation [13, 20, 29] based on the graph neural network (GNN) architecture to capture the high-order connectivity between entities. We also extract features of user attributes to consider user-side information. The experiments on the real dataset verify the effectiveness of HMKR.

The main contributions of this work are summarized as follows:

- We highlight the significance of attribute triples and high-order connectivity of KG for knowledge graph embedding and recommendation.
- We process the user attributes to provide more reliable recommendation.
- We propose the method HMKR, which conducts multi-task training to improve recommendation with the enhanced knowledge graph embedding integrating the attribute information and high-order information.
- We conduct experiments on the real dataset, and HMKR is proved to be effective.

2 RELATED WORK

2.1 Embedding-based Methods for Recommendation

In recent years, embedding has gained many application and development [3]. Most embedding-based methods [36], [10], [24], [22], [30], [2], [12], [28], [31] use various forms of item side information to construct KGs so that the presentation of items can be enriched. And this information can be utilized to build user representation accurately. Some models [34], [16], [1], [21], [5] construct user preferences directly by drawing users into graph to build user-item interaction. Our work can be regarded as a especial embedding-based methods for recommendation.

2.2 Graph Neural Network for Recommendation

Our work applies the idea of GNN. GNN is increasingly utilized for recommendation. It improves the model from the traditional convolution network. The recommender systems based on GNN are mainly collaborative filtering. The specific problems include Web-Scale Recommender Systems [32], GC-MC [19] and so on [26], [9]. In later research, the collaborative filtering applies secondary information. It builds models from two perspectives, one is Social Recommendation [6], [7], the other one is Knowledge-graph-aware Recommendation [23], [25].

3 OUR APPROACH

As shown in Figure 1, HMKR consists of four sub-modules: recommendation module, enhanced knowledge graph embedding module, cross-compress unit and user feature extraction unit.

3.1 User Feature Extraction Unit

Inspired by the existing work [27], we process all user attributes by multi-layer perceptron to output the user feature vectors.

We use $x : \{\mathrm{x}_1, \mathrm{x}_2 \ldots \mathrm{x}_\mathrm{m}\}$ to express the attributes of the user, where x_i means the user attributes. We input user attributes into the user feature extraction unit to obtain the user attributive feature vector X:

$$X = f\left(w_1 x + b_1\right) \tag{1}$$

Where w_1 represents the weight, b_1 represents the bias, and $f(\cdot)$ means the activation function.

We get the embedding of the user after the full connection layer:

$$u = \text{concatenate}(X) \tag{2}$$

3.2 Recommendation Module

The module input is the feature vector $\mathbf{u}$ of user u and the feature vector $\mathbf{v}$ of the item v. Vector $\mathbf{u}$ is the output result of the user feature extraction unit. The vector $\mathbf{v}$ is the original feature of item.

In recommendation module, we use L-layers multi-layer perceptron to extract the user feature:

$$u_L = M(M(\cdots M(u))) = M_L(u) \tag{3}$$

Formula 3 is the extraction of u's feature, where $\mathrm{M}(\mathrm{x}) = \sigma(\mathrm{Wx} + \mathrm{b})$ is a fully-connected neural network layer. W is weight, b is bias, and $\sigma(\cdot)$ is nonlinear activation function.

We extract the feature of item v by applying the cross-compress unit:

$$vL = E_{e \sim S(v)}\left[Cross^L[v]\right] \tag{4}$$

Where $S(v)$ is the associated entity set of item v. More details of cross and compress will be presented in section 3.4.

We predict the probability result of user u participating in item v by using the prediction function after obtaining the latent features of user u and item v:

$$\hat{y}_{\mathrm{uv}} = \sigma f_{\mathrm{RS}}\left(M_L(\text{ concatenate }(X)), vL\right) \tag{5}$$

3.3 Enhanced Knowledge Graph Embedding Module

The module encodes attribute triples and captures the high-order connectivity between entities.

3.3.1 Attribute Embedding. We encode the words or sentences of attribute triples as vectors of fixed length. After studying, we apply LSTM networks to encode attributes into vectors which can conquer the defect of Bag-of-Word encoder [15].

$$\mathbf{a} = f_{lstm}\left(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\right) \tag{6}$$

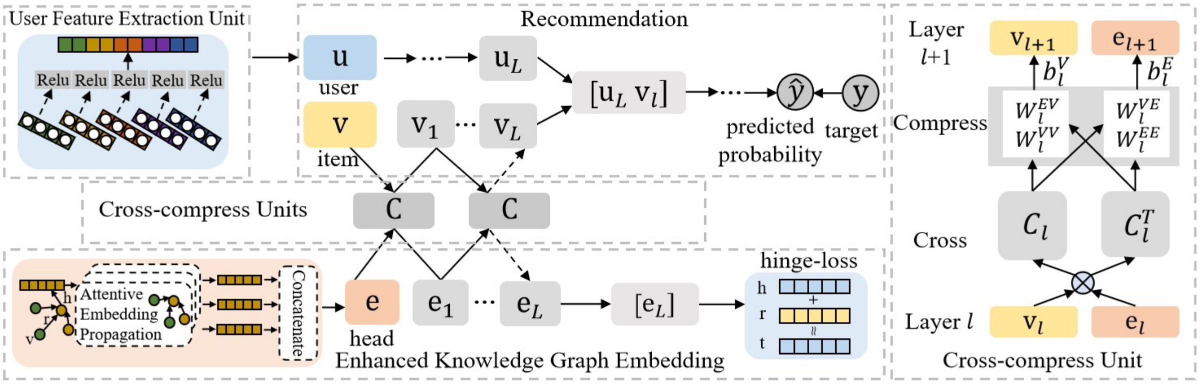


Figure 1: Illustration of the HMKR model.

Where f_{lstm} represents the Long-Short Term Memory network.

3.3.2 Embedding Propagation. Next we conduct embedding propagation based on the architecture of graph convolution network and the idea of graph attention network. Our embedding propagation consists of attentive embedding propagation and embedding aggregation.

The input of our embedding propagation is a set of entities, relations and attribute value embeddings. We employ $\mathbf{h} \in R^k$ to express the embedding of entity h. The neighbor of h is defined as $\mathcal{N}_h = \{t, a \mid (h, r, t) \in T_R \cup (h, r, a) \in T_A\}$. We obtain the vector $\mathbf{H}$ as the improved embedding of entity h through attentive embedding:

$$\mathbf{H} = \sum_{t \in \mathcal{N}_h} \pi(h, r, t)\mathbf{W}(\mathbf{r} + \mathbf{t}) \tag{7}$$

Where $\mathbf{W} \in R^{k' \times k}$ is one learnable linear transformation, $\pi(h, r, t)$ means the significance of entity t to entity h.

We use softmax function on the triples connected with h:

$$\pi(h, r, t) = \frac{\exp(\pi(h, r, t))}{\sum_{t' \in \mathcal{N}_h} \exp\left(\pi\left(h, r', t'\right)\right)} \tag{8}$$

$\pi(h, r, t)$ is realized by a simply feedforward neural network:

$$\pi(h, r, t) = \text{LeakyRelu}\left((\mathbf{W}\mathbf{h})^T \mathbf{W}(\mathbf{r} + \mathbf{t})\right) \tag{9}$$

We use concatenation aggregator to concatenate all embeddings of multi-head graph attention for more stable learning and transform it through LeakyReLu:

$$\mathbf{H}' = \text{LeakyReLu}\left(\mathbf{W}\left(\|_{i=1}^{m} \sum_{t \in \mathcal{N}_h} \pi(h, r, t)^i \mathbf{W}^i(\mathbf{r} + \mathbf{t})\right)\right) \tag{10}$$

Where $\|$ represents concatenation, $\pi(h, r, t)^i$ is i-th normalized attention coefficient and $\mathbf{W}^i$ denotes the linear transformation.

We utilize embedding propagation layers to gather the deeper information of the neighbors so that we can obtain the high-order connectivity information in KGs. The l-th layer embedding of entity h is stated as follows:

$$\mathbf{H}^{(l)} = \sum_{t \in \mathcal{N}_h} \pi(h, r, t)\mathbf{W}\left(\mathbf{r}^{(l-1)} + \mathbf{t}^{(l-1)}\right) \tag{11}$$

After the propagation and aggregation, we extract the feature of entity h by applying the cross-compress unit:

$$\mathbf{e}_L = \mathbb{E}_{v \sim \mathcal{S}(h)}\left[Cross^L[\mathbf{e}]\right] \tag{12}$$

Where $\mathcal{S}(h)$ is the set of relevant items of entity h.

3.4 Cross-compress Unit

The cross-compress unit consists of two operations, cross and compress. The unit interacts the items in the recommendation with the entities in the KG so that features can be shared.

We construct cross feature matrix C_l of layer l. This is the cross operation.

$$C_l = v_l e_l^T = \begin{bmatrix} v_l^{(1)} e_l^{(1)} & \cdots & v_l^{(1)} e_l^{(d)} \\ \vdots & & \vdots \\ v_l^{(d)} e_l^{(1)} & \cdots & v_l^{(d)} e_l^{(d)} \end{bmatrix} \tag{13}$$

Where $\mathbf{v}_l \in R^d$, $\mathbf{e}_l \in R^d$ and d is the dimension of hidden layers.

Follwing the cross operation, we conduct compress operation to get the next layer feature vectors. We project the cross feature matrix into vectors of items and entities in their latent representation spaces:

$$\begin{aligned} \mathbf{v}_{l+1} &= \mathbf{v}_l \mathbf{e}_l^\top \mathbf{w}_l^{VV} + \mathbf{e}_l \mathbf{v}_l^\top \mathbf{w}_l^{EV} + \mathbf{b}_l^V \\ \mathbf{e}_{l+1} &= \mathbf{v}_l \mathbf{e}_l^\top \mathbf{w}_l^{VE} + \mathbf{e}_l \mathbf{v}_l^\top \mathbf{w}_l^{EE} + \mathbf{b}_l^E \end{aligned} \tag{14}$$

Where $w_l \in R^d$ is weight and $b_l \in R^d$ is bias.

Cross-compress units help both tasks adjust and learn automatically.

3.5 Optimization

The loss of HMKR is as follows:

$$\mathcal{L} = \mathcal{L}_{RS} + \mathcal{L}_{KG} + \lambda_2 \|W\|_2^2 \tag{15}$$

$$\mathcal{L}_{RS} = \sum_{u \in \mathcal{U}, v \in \mathcal{V}} \mathcal{J}\left(\hat{y}_{uv}, y_{uv}\right) \tag{16}$$

$$\mathcal{L}_{KG} = \sum_{(h,r,e) \in T} \sum_{(h',r,e') \in T'} \left[\gamma + d(h + r, e) - d\left(h' + r, e'\right)\right]_+ \tag{17}$$

Where $\mathcal{L}_{RS}$ is the loss of recommendation, $\mathcal{L}_{KG}$ is the loss of enhanced KGE, $\lambda_2\|W\|_2^2$ is the regularization term for preventing overfitting.

4 EXPERIMENTS

4.1 Datasets

We utilize MovieLens dataset in our experiments. MovieLens-1M is a dataset widely applied in recommendation. The set has approximately 1 million explicit ratings and contains 6036 users, 2347 items, 753772 ratings and 20195 knowledge graph triples.

4.2 Baselines

We make comparisons with the following four baselines. The settings of the various models remain unchanged.

- **MKR** [24] is one of the inspiration ideas of our method. In the experiment, the inner product is f_{RS}. High-level layer is $K = 1$ and $\lambda_1 = 0.5, \lambda_2 = 10^{-6}, L = 1, d = 8$, t = 3.
- **Wide&Deep** [4] is a deep recommendation method that combines wide and deep channels. In the experiment, the dimensions of user, item and entity are 64, the dimension of deep channel is 100, and the dimension of wide channel is 50.
- **DKN** [22] combines entity embedding and word embedding as multiple channels for prediction. In the experiment, the dimensions of word and entity embedding are 64, and the filters with different window sizes are 128.
- **SI-MKR** [27] is an alternate training recommendation model based on MKR that extracts valuable external information. In the experiment, $\lambda_2 = 10^{-8}$, learning rate of recommendation and embedding are $2e - 4$ and $2e - 5$.

4.3 Experiments setup

The training, validation and test sets in our HMKR is 6:2:2. We conduct every experiment 3 times and take the average result. We evaluate HMKR in two experimental scenarios. One is the click-through rate (CTR) prediction in which we use AUC and ACC to evaluate the result. Another scenario is the top-K recommendation in which we evaluate the result through the set of Precision@K and Recall@K.

4.4 Performance

Table 1 is the AUC and ACC of HMKR and other baselines in CTR prediction. Figure 2 is the result of Recall@K and Precision@K in top-K recommendation.

We can observe that Wide&Deep splices the attributes simply, so the performance is ordinary. DKN performs poorly because it is more suitable for long text recommendation such as news recommendation. SI-MKR performs better than MKR as it considers user attributes, item text and multi-value attributes based on MKR. Our method performs best because we consider the high-order connectivity of entities, attribute triples with rich semantics and user attributes.

Table 2 is the performance of different models in different ratios of the training set. We test the proportions of the training set as

Table 1: Results of AUC and ACC in CTR prediction.

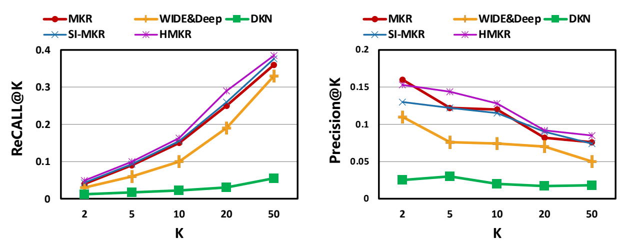
Model	AUC	ACC
MKR	0.917	0.843
Wide&Deep	0.898	0.820
DKN	0.655	0.589
SI-MKR	0.921	0.845
HMKR	0.924	0.847

Table 2: Results of AUC in CTR prediction with different ratios of training set r.

Model	20%	40%	60%	80%	100%
MKR	0.874	0.882	0.897	0.908	0.917
Wide&Deep	0.802	0.815	0.840	0.876	0.898
DKN	0.582	0.601	0.620	0.638	0.655
SI-MKR	0.876	0.884	0.898	0.900	0.921
HMKR	0.877	0.886	0.899	0.910	0.923

100%, 80%, 60%, 40%, 20% respectively to explore the performance in sparse scenarios.

From the experimental results, we can know that our model further relieves the sparsity problem and improve the effectiveness of the recommender systems on the basis of existing research.



(a) Recall@K (b) Precision@K

Figure 2: The results of Recall@K and Precision@K in top-K recommendation.

5 CONCLUSIONS

In this work, we propose HMKR. Our model takes attribute triples that encode rich semantic information into consideration, uses the idea of GNN to capture high-order connections between knowledge graph entities and takes user-side information into account. HMKR shares high-order features of items and entities, and alternately trains recommendation and knowledge graph embedding. The user feature extraction and enhanced knowledge graph embedding further relieve the sparsity and cold start problems of the recommendation. Experiments on the MovieLens dataset prove that the HMKR model has reached better performance than other models.

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation. *Algorithms* 11 (2018), 137.

[2] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences. *The World Wide Web Conference* (2019).

[3] Liang Chang, Manli Zhu, Tianlong Gu, Chenzhong Bin, Junyan Qian, and Ji Zhang. 2017. Knowledge Graph Embedding by Dynamic Translation. *IEEE Access* 5 (2017), 20898–20907.

[4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishikesh B. Aradhye, Glen Anderson, Gregory S. Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. WideDeep Learning for Recommender Systems. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (2016).

[5] Amine Dadoun, Raphael Troncy, Olivier Ratier, and Riccardo Petitti. 2019. Location Embeddings for Next Trip Recommendation. *Companion Proceedings of The 2019 World Wide Web Conference* (2019).

[6] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Yihong Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. *The World Wide Web Conference* (2019).

[7] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2020. A Graph Neural Network Framework for Social Recommendations. *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1.

[8] Q. Guo, F. Zhuang, C. Qin, H Zhu, X. Xie, H Xiong, and Q. He. 2020. A Survey on Knowledge Graph-Based Recommender Systems. *Scientia Sinica Informationis* 50, 7 (2020), 937.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yaliang Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).

[10] Jin Huang, Wayne Xin Zhao, Hong-Jian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018).

[11] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu. 2020. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. (2020).

[12] Kevin Joseph and Hui Jiang. 2019. Content based News Recommendation via Shortest Entity Distance over Knowledge Graphs. *Companion Proceedings of The 2019 World Wide Web Conference* (2019).

[13] Thomas Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv* abs/1609.02907 (2017).

[14] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[15] WenQian Liu, HongYun Cai, Xu Cheng, Sifa Xie, Yipeng Yu, and Hanyu Zhang. 2019. Learning High-order Structural and Attribute information by Knowledge Graph Attention Networks for Enhancing Knowledge Graph Embedding. *ArXiv* abs/1910.03891 (2019).

[16] Enrico Palumbo, Giuseppe Rizzo, and Raphael Troncy. 2017. entity2rec: Learning User-Item Relatedness from Knowledge Graphs for Top-N Item Recommendation. *Proceedings of the Eleventh ACM Conference on Recommender Systems* (2017).

[17] Tmk Xiaoyuan Su. [n.d.]. Review Article A Survey of Collaborative Filtering Techniques. ([n. d.]).

[18] Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding. In *SEMWEB*.

[19] Rianne van den Berg, Thomas Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *ArXiv* abs/1706.02263 (2017).

[20] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. 2018. Graph Attention Networks. *ArXiv* abs/1710.10903 (2018).

[21] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. 2018. SHINE: Signed Heterogeneous Information Network Embedding for Sentiment Link Prediction. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (2018).

[22] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. *Proceedings of the 2018 World Wide Web Conference* (2018).

[23] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge Graph Convolutional Networks for Recommender Systems with Label Smoothness Regularization. *ArXiv* abs/1905.04413 (2019).

[24] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. *The World Wide Web Conference* (2019).

[25] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019).

[26] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019).

[27] Yuequn Wang, Li yan Dong, Henghai Zhang, Xintao Ma, Yongli Li, and Minghui Sun. 2020. An Enhanced Multi-Modal Recommendation Based on Alternate Training With Knowledge Graph Representation. *IEEE Access* 8 (2020), 213012–213026.

[28] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon M. Jose. 2019. Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019).

[29] Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. 2020. Inductive Representation Learning on Temporal Graphs. *ArXiv* abs/2002.07962 (2020).

[30] Deqing Yang, Zikai Guo, Ziyi Wang, Juyang Jiang, Yanghua Xiao, and W. Wang. 2018. A Knowledge-Enhanced Deep Recommendation Framework Incorporating GAN-Based Models. *2018 IEEE International Conference on Data Mining (ICDM)* (2018), 1368–1373.

[31] Yuting Ye, Xuwu Wang, Jiangchao Yao, Kunyang Jia, Jingren Zhou, Yanghua Xiao, and Hongxia Yang. 2019. Bayes EMbedding (BEM): Refining Representation by Integrating Knowledge Graphs and Behavior-specific Networks. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019).

[32] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018).

[33] Ji Zhang, Leonard Tan, Xiaohui Tao, Dianwei Wang, Jia-Ching Ying, and Xin Wang. 2019. Learning Relational Fractals for Deep Knowledge Graph Embedding in Online Social Networks. In *WISE*.

[34] Yongfeng Zhang, Qingyao Ai, Xu Chen, and Pengfei Wang. 2018. Learning over Knowledge-Base Embeddings for Recommendation. *ArXiv* abs/1803.06540 (2018).

[35] Yu Lin Zhang and Qiang Yang. 2017. A Survey on Multi-Task Learning. *ArXiv* abs/1707.08114 (2017).

[36] S. A. Zhu, A Qg, Y. B. Jie, F. C. Hui, D Gg, Z. A. Jie, and E Rb. [n.d.]. Research commentary on recommendations with side information: A survey and research directions. *Electronic Commerce Research and Applications* 37 ([n. d.]).