2021 IEEE 37th International Conference on Data Engineering (ICDE) | 978-1-7281-9184-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICDE51399.2021.00291

Improving Conversational Recommender System by Pretraining Billion-scale Knowledge Graph

Chi-Man Wong[‡†],Fan Feng[‡], Wen Zhang[§], Chi-Man Vong[†], Hui Chen[‡], Yichi Zhang[‡], Peng He[‡],
Huan Chen[*], Kun Zhao[‡], Huajun Chen[§¶]
[‡*]Alibaba Group, [†]University of Macau, [§]Zhejiang University, China
[‡]{chiman.wcm, fengfan.fengfan, weidu.ch, yichi.zyc, hepeng.hp, kun.zhao}@alibaba-inc.com
[*]{shiwan.ch}@taobao.com
[§]{wenzhang2015, huajunsir}@zju.edu.cn

***Abstract*—Conversational Recommender Systems (CRSs) in E-commerce platforms aim to recommend items to users via multiple conversational interactions. Click-through rate (CTR) prediction models are commonly used for ranking candidate items. However, most CRSs are suffer from the problem of data scarcity and sparseness. To address this issue, we propose a novel knowledge-enhanced deep cross network (K-DCN), a two-step (pretrain and fine-tune) CTR prediction model to recommend items. We first construct a billion-scale conversation knowledge graph (CKG) from information about users, items and converations, and then pretrain CKG by introducing knowledge graph embedding method and graph convolution network to encode semantic and structural information respectively. To make the CTR prediction model sensible of current state of users and the relationship between dialogues and items, we introduce user-state and dialogue-interaction representations based on pre-trained CKG and propose K-DCN. In K-DCN, we fuse the user-state representation, dialogue-interaction representation and other normal feature representations via deep cross network, which will give the rank of candidate items to be recommended. We experimentally prove that our proposal significantly outperforms baselines and show it's real application in Alime.**

***Index Terms*—Conversational Recommender Systems, Knowledge Graph, Entity Representation, CTR.**

I. Introduction

Conversational Recommender Systems (CRS) [1]–[5] are commonly used to improve customer experience on E-commerce platforms. Many online stores have customer service chatbots to help users find their ideal items, which contributes to the revenue greatly. The goal of these chatbots is to identify users' intentions by multiple conversational interactions, and then recommend a list of most suitable items to them. As shown in figure 1, in a certain dialogue, when a user asks for recommendation, the CRS employs a three-stage pipeline. First, it makes semantic analysis of the query, then gets a list of candidate items, and finally ranks candidate items through deep click-through rate (CTR) models.

In this paper, we focus on the ranking process of CRS, thus the key task is to estimate the CTR. CTR prediction models [6]–[8] with various deep networks have been proposed and achieved good performance. However, they mainly rely on abundant behavioral records of users on items, while these records might be very sparse because of limited interactions with a few chatbots among millions of online shops, which would lead to the over-fitting problem.

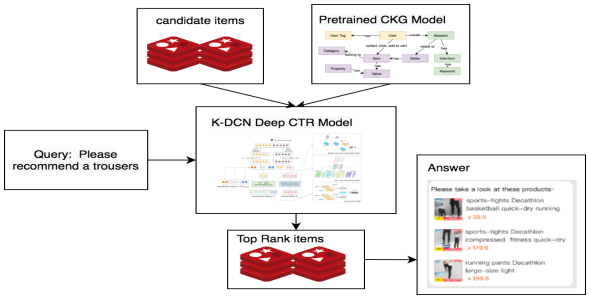


Fig. 1. The workflow of K-DCN.

To solve the data scarcity and sparseness problem, we propose to make user, item and conversation information into consideration, since certain preference for items might be expressed in conversation.

To better organize and utilize information, our proposal mainly includes two parts: 1) construct and pre-train of conversational knowledge graph (CKG); and 2) knowledge-enhanced deep cross network (K-DCN) is employed for fine-tuning. Firstly, we construct a billion-scale CKG from information about user, item and conversation, and then is employed to learn a pre-trained model to obtain better entity representations. There are two common approaches to learn entity representations in KGs. The one is Graph Neural Network(GNN) based methods [9]–[13] like GCN [9], which models structure information of an entity by aggregating all its neighbors' information. The other one is knowledge graph emebdding (KGE) methods like translation-based methods [14]–[22], which are good at capture semantic information of entities. Since both structure and semantic information of entities are important, we combine GNN-based method and KGE method together for pre-training CKG via joint learning. In the second part, we propose a novel two-step ranking model, K-DCN, in which we introduce user-state representations and dialogue-interaction representations based on pre-trained CKG, to help the CTR prediction model sensible of the uses' state and relationships

¶Corresponding author.

978-1-7281-9184-3/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDE51399.2021.00291

between dialogues and items. We experiment our proposal on real-life e-commerce CRS. The results show that our proposal outperforms baselines and converge faster during training. Finally, we also show the application of our proposal in Alimi, a chatbot in Taobao platform.

In summary, our contributions are:

- We propose to integrate both user, item and conversation information in the form of conversation knowledge graph to solve the data scarcity and sparseness problem of CTR prediction models in CRS.
- We introduce a novel method to organize and utilize various information, via first get good entity representations from billion-scale pre-trained CKG, and then fine-tune K-DCN with user-state representations and dialogue-interaction representation introduced.
- We experimentally show that our proposal outperforms baselines and converge faster during training on real-life e-commerce datasets, and we also show the real application of our proposal on Alimi.

II. Related Work

A. Translation-based KG embedding

Translation-based methods adopt a scoring function $f(h, r, t)$ to measure the plausibility of a fact (h,r,t) from KG. E.g., in TransE [14] assumes $f(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$. TransH [15] projects entity representation onto relation' hyperplane before calculating score. TransR [16] and TransD [17] project entities from entity space to relation space via projection matrix. TransEdge [18] contextualizes relation by edge-centric embedding. DistMult [19] captures relational semantics by the composition of relation; RotateE [20] learns various relation patterns by modeling relation as rotation. However, all of the above methods do not learn the structure information of KG, which is an important information to provide robustness of the model.

B. GCN-based KG embedding

GCN-based method represents the embedding of each entity by iteratively propagating neighbor information. For example, GCN [9] introduces a first-order approximation of ChebNet to perform graph convolution so that the number of parameters are restrained and the issue of over-fitting is avoided; AGCN [10] could learn all graph structure information by introducing a distance metric learning; DGCN [11]propose a dual graph convolution to encodes both local and global structure information by normalized adjacency and positive pointwise mutual information (PPMI) matrix; GAT [12] employed masked self-attentional layer to learn the importance of neighborhoods' information by assigning different weights to them; GeniePath [21] learns the importance of different sized neighborhoods by using an adaptive path layer, such that both breadth and depth can be explored for information extraction; Gaan [13] control each attention head's importance by sub-convolution network; Although these methods could encode the neighbor's information of an entity, the semantic meaning can not be learned, which would deteriorate the performance.

C. Deep Click-through Rate Prediction model

The development of deep neural networks (DNNs) and embedding techniques provides a better way to learn feature expression in recommender systems. For example, Deep & Cross Network(DCN) [6] and DeepFM [7], [23] contain both deep component and shallow component to capture the feature interaction automatically. In common Click-through Rate Prediction model, the input data with sparse and dense features mainly concern about user action and product attribute, and only use the embedding technique to reduce the dimensionality of categorical features. But in dialog system, the input must be more specific about current status. Inspired by the DKN [8], the knowledge-level embeddings of entities on the dialog system can enrich the use state and dialogue interaction representation directly.

III. Method

Our method contains two parts: 1) constructing a Conversation Knowledge Graph(CKG) and pre-training it to encode structure and semantic information; 2) fine-tuning a ranking model DCN based on normal features and features from pre-trained knowledge graph representations, including user-state representations and dialogue-interaction representations.

A. Constructing and Pre-training of CKG

1) Conversation Knowledge Graph Construction: We make knowledge graph as a way to encode diverse information related to conversational recommendation system and we construct the CKG from real-world scenarios of user-chatbot conversation. There are three kinds of information contained in CKG:

- Information about users: in our platform, each user has a list of tags, such as gender, shopping history, etc, which could help to catch the interest of users. Thus we build triples encoding the tag information in CKG in the form of (user, has, user tag).
- Information about items: our platform contains rich information about items like category and seller of items, and also properties of items such as color, brand, etc. For categories and seller information, we build triple in the form of (item, belong to, category) and (seller, has, item) respectively. For properties and values, two kinds of triples are built (item, has, value) and (property, has, value). This item information could contribute the mapping to users interests greatly.
- Information about conversation: in the chatbot, many sessions are created when user chat with the bot, the intentions and keywords from the conversation could help understand the intention of a user more precisely. Thus,we could build four kinds of triples based on conversation history, (user, created, session), (session, relate to, seller), (session, has, intention) and (intention, has, keywords).

The schema of CKG is shown in figure 3. Finally, CKG contains 600 million entities and 6 billion triples.

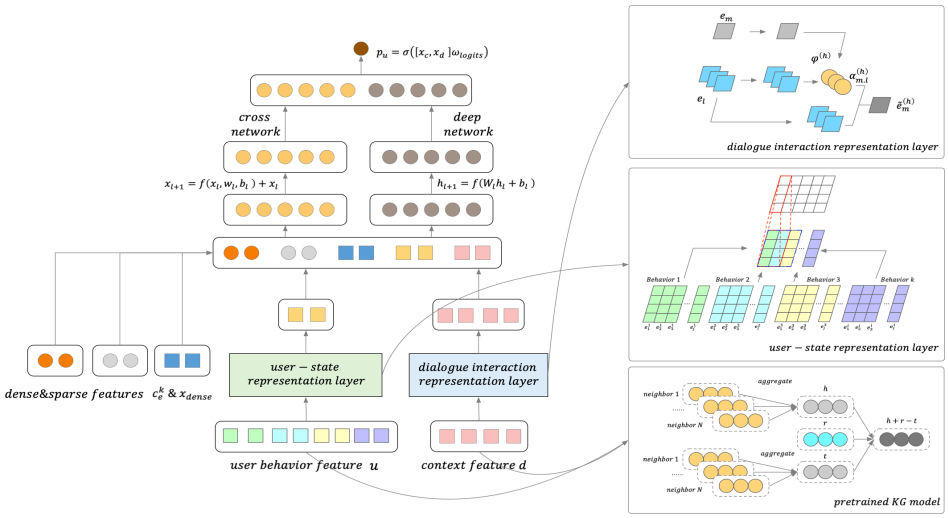


Fig. 2. The architecture of knowledge-enhanced deep cross network (K-DCN).

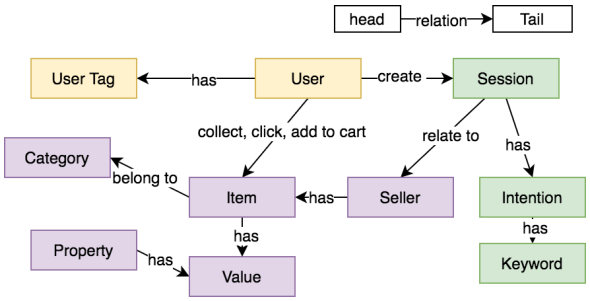


Fig. 3. The conversation knowledge graph.

2) Conversation Knowledge Graph Pre-training: As known that structural and semantic information are both valuable in knowledge graph, thus our CKG pre-training contains two modules, structural embedding module and semantic embedding module to capture them respectively.

Structural Embedding Module: Graph neural network(GNN) has been proved to be useful in encoding structural information [9]–[13]. Thus we employ GCN [9] to learn the structur representation of entities in CKG where a vector representation $\mathbf{e} \in \mathbb{R}^d$ for each entity $e \in \mathcal{K}$. Gathering all entity embedding together in order, we get $\mathbf{E} \in \mathbb{R}^{n_e \times d}$, where n_e is the number of entities and d is the embedding dimension. A multi-layer GCN is given with a simple layer-wise propagation rule and the ith layer could be represented as:

$$X^{(i+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}X^{(i)}W^{(i)}) \tag{1}$$

where $\hat{A} \in \mathbb{R}^{n \times n}$ is adjacency matrix where $\hat{A}_{ij} = 1$ if e_i and e_j are connected in CKG and $\hat{A}_{ij} = 0$ otherwise. $\hat{D}$ is the diagonal degree matrix of $\hat{A}$. n is the number of entities in CKG. σ is the activation function and we make it sigmoid function during experiments. $W^{(i)}$ is the weight matrix in ith layer. The output $X^{(i+1)}$ encodes the structure information from ith layer with $X^{(i)}$ as structure information of previous layers. In the first layer, we make $X^{(1)} = \mathbf{E}$.

Semantic Embedding Module: Many knowledge graph embedding(KGE) methods are proposed to encode the semantic information implicitly. Considering the effectiveness and efficiency, we employ TransE in semantic embedding module. For a triple (e_i, r, e_j), the score function of it is defined as:

$$f(e_i, r, e_j) = ||\mathbf{e}_i + \mathbf{r} - \mathbf{e}_j|| \tag{2}$$

where $\mathbf{e}_i, \mathbf{r}$ and $\mathbf{e}_j$ are embeddings of e_i, r and e_j respectively. With this function, positive triples should get lower scores and negative ones get higher ones.

Pretraining: we jointly train structure embedding module and semantic embedding module to learn entity and relation embeddings. During co-training, we make the output of multi-layer GCN $X^{(i+1)}$ as the entity embedding matrix for TransE which means $\mathbf{e}_k = X_k^{(i+1)} (k \in [0, n-1])$. The training target to minimize is the following margin-based ranking loss:

$$L = \sum_{(e_i, r, e_j) \in \mathcal{T}} [f(e_i, r, e_j) + \gamma - f(e_i', r', e_j')]_+ \tag{3}$$

where (e_i', r', e_j') is the negative sample of (e_i, r, e_j) by randomly replace e_i or e_j with $e \in CKG$. Function $[x]_+ = 0$ if $x < 0$, otherwise $[x]_+ = x$. After construction and pre-training of CKG, the trained entity embeddings will be used to fine-tune CTR prediction models to improve conversational recommendation system.

B. K-DCN for Fine-tuning

Normally, common sparse and dense features, eg. statistical features, are used in CTR prediction models, while in conversation recommendation system, user's immediate question to chatbot is also important for understanding user's intention, thus we consider and construct user-state and dialogue-interaction representations based on pre-trained CKG as additional features.

1) User-state Representation: Users' state could be captured by their previous behaviors. For one user, suppose he/she has k behaviors $\mathcal{B} = \{b_i | b_i = \{e_1^{(i)}, e_2^{(i)}, ...\}, i \in [1, k]\}$ where b_i is an user-item click sequence. For each b_i, we first average the embedding of each item $e \in b_i$ from pre-trained CKG and get a behavior vector $\mathbf{b}_i$:

$$\mathbf{b}_i = \frac{1}{|b_i|} \sum_{e_j^{(i)} \in b_i} \mathbf{e}_j^{(i)} \tag{4}$$

Then we vertically concatenate all behavior vectors as $\mathbf{B} \in \mathbb{R}^{d \times k}$ as shown in Figure (2) and employ convolutional neural networks(CNN) [24] on $\mathbf{B}$ to model the local information of the user-state representations:

$$\mathbf{u} = f_{CNN}(\mathbf{B} * \mathbf{w} + b) \tag{5}$$

where $*$ is is the convolution operator, and $b \in \mathbb{R}$ is a bias. We show the details of this step in right side of Figure (2) marked as user-state representation layer.

2) Dialogue-interaction Representation: To build dialogue interactions, we first extract a set of keywords from users question $\mathcal{W}_{query} = \{w_q^{(1)}, w_q^{(2)}, ..., w_q^{(m)}\}$. Given a candidate item c, we also extract a set of keywords from its title $\mathcal{W}_{title} = \{w_t^{(1)}, w_t^{(2)}, ..., w_t^{(n)}\}$. Then, we gather all keywords together and get their representations from pre-trained CKG's entity embeddings, represented as $\mathbf{W} = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, ..., \mathbf{w}^{(m+n)}\}$. Then we fed keyword embeddings into a multi-head self attention network [25] to model the inter-relationship between query and candidate item. Formally:

$$a_{ij} = \frac{exp((\mathbf{M}_a \mathbf{w}_i) \circ (\mathbf{M}_b \mathbf{w}_j)))}{\sum_{t=1}^{m+n} exp((\mathbf{M}_a \mathbf{w}_i) \circ (\mathbf{M}_b \mathbf{w}_k))} \tag{6}$$

where $\circ$ is inner-product operation and $\mathbf{W}_a \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_b \in \mathbb{R}^{d \times d}$ are matrices transforming input embeddings to a new space. where $\psi(\mathbf{w}_i, \mathbf{w}_j) = innerproduct(M_a \mathbf{w}_i, M_b \mathbf{w}_j)$ is the attention function.M_a, M_b are matrix which

Next, we update the representation through a weighted sum according to a_{ij}:

$$\widetilde{\mathbf{w}}^{(i)} = \sum_{j=1}^{m+n} a_{i,j} (\mathbf{W}_v \mathbf{e}^{(i)}) \tag{7}$$

Finally, we concatenate all updated word embeddings as the dialogue-interaction representation $\mathbf{d} \in \mathbb{R}^{(m+n) \times d}$:

$$\mathbf{d} = [\widetilde{\mathbf{w}}^{(1)}, \widetilde{\mathbf{w}}^{(2)}, ..., \widetilde{\mathbf{w}}^{(m+n)}] \tag{8}$$

3) K-DCN Model: Inputs of K-DCN are various feature representations for a candidate item e. Besides aforementioned user-state and dialogue interaction representations $\mathbf{u}$ and $\mathbf{d}$, other common features are also considered, including categorical features and statistical features. First, we concatenate all feature vectors as follows:

$$\mathbf{f} = [\mathbf{c}_e^{(1)}, \mathbf{c}_e^{(2)}, ..., \mathbf{c}_e^{(k)}, \mathbf{x}_{dense}, \mathbf{u}, \mathbf{d}] \tag{9}$$

where $\mathbf{c}_e^{(1)}, \mathbf{c}_e^{(2)}, ..., \mathbf{c}_e^{k}$ are embeddings of categories that target item e belongs to, which will be randomly initialized during training. $\mathbf{x}_{dense}$ is statistical feature vector of e where values could be price, sales etc.

With the feature representation $\mathbf{f}$ for e, we feed it into the cross network and deep network respectively which are two common components in CTR prediction model. The cross network is composed of n_c cross layers and the ith layer could be represented as:

$$\mathbf{x}_c^{(i+1)} = \mathbf{f}(\mathbf{x}_c^{(i)})^T \mathbf{w}_i + \mathbf{b}_i + \mathbf{x}_c^{(i)} \tag{10}$$

where $\mathbf{x}_c^0 = \mathbf{f}$. The deep network is a n_d layers of fully-connected feed-forward network and the function of the ith layer is

$$\mathbf{x}_d^{(i+1)} = \sigma\left(W_i \mathbf{x}_d^{(i)} + \mathbf{b}_l\right) \tag{11}$$

Then we concatenate the two outputs from the above two networks and feed the concatenated vector into a standard logits layer. Formally:

$$p = \sigma\left(\left[\mathbf{x}_c^{(n_c)}, \mathbf{x}_d^{(n_d)}\right] \mathbf{W}_{\text{logits}}\right) \tag{12}$$

To train K-DCN, we minimize the following log likelihood loss function:

$$L_{K-DCN} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \tag{13}$$

where N is the number of input samples. $y_i \in \mathbb{R}^{n_{cand} \times 1}$ is the recommendation label of the ith sample and n_{cand} is the number of candidate items to be recommended.

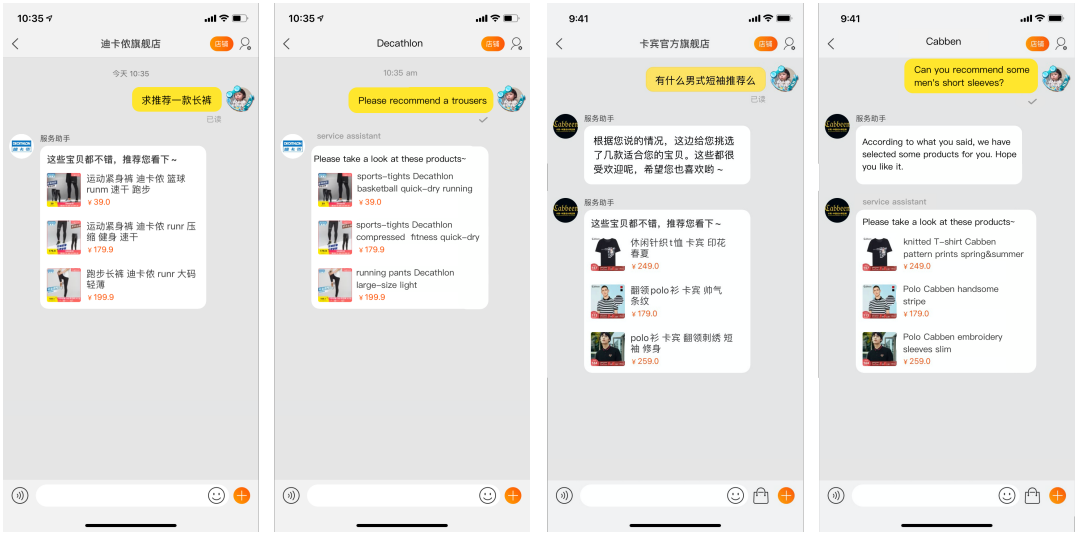
IV. Experiment and Discussion

A. Datasets

K-DCN is pretrained and fine-tuned on a private dataset from scratch. The dataset for training CTR prediction model is from the conversation of Alime salebot (As illustrated in Figure 1). As shown in table I,We have sampled 900K records from 10 categories.

B. Implementation Details

- CKG. The pre-trained KG model is trained on billion-scale converation knowledge graph (CKG). CKG has a total number of 500M entities and 6B triplets. We processd the data into triplets using Alibaba max-reduce framework (called Maxcompute). We removed the attributes with occurrences that are less than 5000 in CKG. Such attributues are very sparse that are likely to deteriorate the model performance. For trainging, we employed the tool Graph-learn from [26] to perform sampling of 10 neighbors for structure pretrain, and the batch size is 512. Adam optimizer is employed with initial learning rate = 0.0001; each training batch size = 1000, and node embedding size = 64. The model is trained with 50 parameter servers and 200 GPUs for 5 epochs. The whole training consumed 2 days.
- K-DCN. The K-DCN is implementated on Tensorflow. The hyperparameters are tuned with grid search. The optimal hyperparameter settings were 2 deep layers of



(a) example A (b) example B

Fig. 4. Two Real-word examples of conversation from Taobao with English translation on the right.

size 512 and 4 cross layers for the K-DCN model. More specifically, we use 4 kinds of behaviors in user state representation and the size of the convolutional operator is $N \times itemEmbeddingSize$, where N is 2 and 4. And the attention head number for text embedding is 4. For different size of convolutional operator, the outputs are concated as final output. We use mini-batch stochastic optimization with Adam [27] optimizer and the batch size is set at 512.

C. Results and Discussion

We compare our methods with 5 baselines: Deep & Cross Network (DCN) [6] , Wide&Deep Network (WDN) [28], Deep Neural Network (DNN) [29], Gradient Boosted Decision Trees (GBDT) [30], and Logistic Regression (LR) [31] in 10 datasets, and the results are shown in II. The proposed K-DCN outperforms all the baselines in all datasets.

Specifically, K-DCN achieves a higher score of 2.6% for the digital accessories category. The main reason is that the number of training data is very limited for digital accessories category, which verifies our model which was pretrained from conversation knowledge can alleviate the issue of data scarcity. For all datasets, K-DCN has an average improvement of 1.2%, which verifies that the proposed KG embeddings from conversation knowledge graph can provide meaningful and useful structure and semantic information to CTR prediction model. What's more, as shown in Figure 5, it would also accelerate the convergence when the pretrained KG embedding were used to initialize the CTR prediction model. We have also tested our K-DCN with DCN in real application in Alime (as shown in Figure. 4. From table III, K-DCN has relative improvement of 4.2% over DCN, which proved the effectiveness of K-DCN.

TABLE I
THE NUMBER OF TRAINING AND TESTING DATA FOR DEEP CTR PREDICTION MODEL.

Category	# Train	# Test
Beauty Skin Care/Body Care/Essential Oil(BS)	157027	16028
Women's wear(MW)	139167	14205
Sports shoes(SS)	104182	10634
Women's and Men's underwear/Home wear(WN)	98705	10075
Sports clothes/Casual wear(SC)	93180	9511
Bed linings (BL)	87537	8935
Cleanser/Sanitary Napkin/Paper/Aromatherapy(CS)	58410	5962
Men's clothes(MC)	57263	5845
Digital accessories(DA)	49455	5048
Kitchen appliances(KA)	55069	5621

TABLE II
K-DCN AND 5 BASELINES' PERFORMANCE IN 10 DATASETS.

Category	AUC					
	K-DCN	DCN	WDL	DNN	GBDT	LR
BC	**0.617**	0.608	0.6	0.598	0.584	0.58
WM	**0.652**	0.636	0.63	0.633	0.611	0.592
SS	**0.658**	0.652	0.65	0.649	0.619	0.603
WN	**0.663**	0.65	0.647	0.646	0.616	0.594
SC	**0.667**	0.66	0.653	0.65	0.618	0.604
BL	**0.631**	0.62	0.61	0.604	0.598	0.574
CS	**0.625**	0.624	0.622	0.621	0.61	0.602
MC	**0.647**	0.63	0.628	0.626	0.603	0.574
DA	**0.706**	0.68	0.677	0.671	0.675	0.646
KA	**0.654**	0.64	0.642	0.64	0.638	0.622
Average	**0.652**	0.64	0.636	0.633	0.617	0.59

TABLE III
THE ONLINE AB TEST IN ALIME.

Platform	% Improvement of K-DCN over DCN
Alime	4.2%

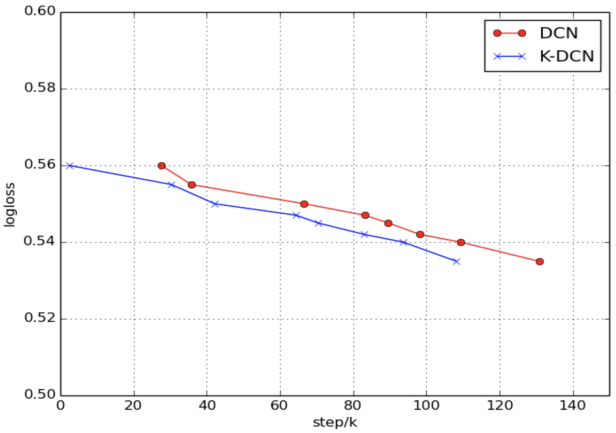


Fig. 5. Convergence speed comparison between K-DCN and DCN.

V. Conclusion

In this paper, we put forward a two-step CTR prediction model K-DCN. We first propose a pretrained model KGM based on a billions scale conversation knowledge graph. Then we propose a knowledge-enhanced deep cross network (K-DCN) model based on KGM as well as some dense and sparse features. Experimental results show that our model obtains substantial performance on 10 datasets compared to baselines. K-DCN have been applied to a real conversational chatbot in one of the largest E-commerce company. In future work, we plan to design a learning framework which can uniformly exploit evidence from heterogeneous knowledge sources, such as entity descriptions and knowledge schema.

VI. Acknowledgements

This work is funded by NSFC91846204/U19B2027/61473260, national key research program 2018YFB1402800 and supported by Alibaba Group through Alibaba Innovative Research Program.

References

[1] J. Gao, M. Galley, and L. Li, "Neural approaches to conversational ai," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 1371–1374.

[2] A. Rahman, A. Al Mamun, and A. Islam, "Programming challenges of chatbot: Current and future prospective," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. IEEE, 2017, pp. 75–78.

[3] A. Fadhil, "Can a chatbot determine my diet?: Addressing challenges of chatbot application for meal recommendation," *arXiv preprint arXiv:1802.09100*, 2018.

[4] I. Nica, O. A. Tazl, and F. Wotawa, "Chatbot-based tourist recommendations using model-based reasoning." in *ConfWS*, 2018, pp. 25–30.

[5] C. Hildebrand and A. Bergner, "Ai-driven sales automation: Using chatbots to boost sales," *NIM Marketing Intelligence Review*, vol. 11, no. 2, pp. 36–41, 2019.

[6] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*, 2017, pp. 1–7.

[7] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," *arXiv preprint arXiv:1703.04247*, 2017.

[8] H. Wang, F. Zhang, X. Xie, and M. Guo, "Dkn: Deep knowledge-aware network for news recommendation," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1835–1844.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[10] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[11] C. Zhuang and Q. Ma, "Dual graph convolutional networks for graph-based semi-supervised classification," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 499–508.

[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[13] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv preprint arXiv:1803.07294*, 2018.

[14] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in neural information processing systems*, 2013, pp. 2787–2795.

[15] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.

[16] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[17] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 687–696.

[18] Z. Sun, J. Huang, W. Hu, M. Chen, L. Guo, and Y. Qu, "Transedge: Translating relation-contextualized embeddings for knowledge graphs," in *International Semantic Web Conference*. Springer, 2019, pp. 612–629.

[19] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[20] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[21] Z. Liu, C. Chen, L. Li, J. Zhou, X. Li, L. Song, and Y. Qi, "Geniepath: Graph neural networks with adaptive receptive paths," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 4424–4431.

[22] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, "Interaction embeddings for prediction and explanation in knowledge graphs," in *WSDM*. ACM, 2019, pp. 96–104.

[23] S. Wang, "Research of shopping recommendation system based on improved wide-depth network," *MS&E*, vol. 768, no. 7, p. 072072, 2020.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[26] R. Zhu, K. Zhao, H. Yang, W. Lin, C. Zhou, B. Ai, Y. Li, and J. Zhou, "Aligraph: a comprehensive graph neural network platform," *Proceedings of the VLDB Endowment*, vol. 12, no. 12, pp. 2094–2105, 2019.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[29] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[31] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.