Gated Heterogeneous Graph Representation Learning for Shop Search in E-Commerce

Xichuan Niu[1], Bofang Li[2], Chenliang Li[3†], Rong Xiao[2], Haochuan Sun[2], Honggang Wang[2], Hongbo Deng[2], Zhenzhong Chen[1]
[1]School of Remote Sensing and Engineering, Wuhan University, China
[2]Alibaba Group, Hangzhou, China
[3]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China
[1]{niuxichuan, zzchen}@whu.edu.cn, [2]{bofang.lbf, xiaorong.xr, haochuan.shc, honggang.whg, dhb167148}@alibaba-inc.com, [3]cllee@whu.edu.cn

ABSTRACT

In e-commerce search, vectorized matching is the most important approach besides lexical matching, where learning vector representations for entities (*e.g.,* query, item, shop) plays a crucial role. In this work, we focus on vectorized search matching model for shop search in Taobao[1]. Unlike item search, shop search is faced with serious behavior sparsity and long-tail problem. To tackle this, we take the first step to transfer knowledge from item search, *i.e.,* leveraging items purchased under a query and the shops they belong to. Moreover, we propose a novel **g**ated **h**eterogeneous graph **l**earning model (named GHL) to derive vector representations for entities. Both first-order and second-order proximity of queries and shops are exploited to fully mine the heterogeneous relationships. And to relieve long-tail phenomenon, we devise an innovative gated neighbor aggregation scheme where each type of entities (*i.e.,* hot ones and long-tail ones) can benefit from the heterogeneous graph in an automatic way. Finally, the whole framework is jointly trained in an end-to-end fashion. Offline evaluation results on real-world data of Taobao shop search platform demonstrate that the proposed model outperforms existing graph based methods, and online A/B tests show that it is highly effective and achieves significant CTR improvements.

KEYWORDS

Heterogeneous Graph, Gated Mechanism, E-commerce, Shop Search

ACM Reference Format:
Xichuan Niu, Bofang Li, Chenliang Li, Rong Xiao, Haochuan Sun, Honggang Wang, Hongbo Deng, Zhenzhong Chen. 2020. Gated Heterogeneous Graph Representation Learning for Shop Search in E-Commerce. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3412087

1 INTRODUCTION

Nowadays as online shopping is becoming increasingly indispensable, e-commerce search has become an essential part of people's life. Taobao, one of the world's largest e-commerce platforms, provides plentiful search services to help customers find desired items or shops. Many efforts have been made to improving user's e-commerce search experience both in academic and industrial community.

In this paper, we focus on the task of vectorized search matching in the context of shop search in Taobao, where user queries and shops will be represented as vectors in a latent embedding space. The goal of this task is to estimate click-through rate (CTR) which is calculated by some function (*e.g.,* inner product) over query and shop vectors. One specific challenge is how to obtain a semantic yet robust query and shop representation, which is critical for retrieving satisfactory search results.

A commonly used method is to feed vector representations into multi-layer perceptrons (MLPs) to model the non-linear feature interactions for relevance estimation. Recently, some graph embedding algorithms are introduced into search CTR optimization in a two-stage manner [7, 9]. These methods take advantage of graph embedding pretraining techniques [2, 5] and the learned node representations are further injected into CTR prediction. A latest model GIN [4] uses the end-to-end joint training method of co-occurrence commodity graph learning and CTR prediction, which achieves significant CTR improvements.

Though these approaches attain remarkable performance gains, they cannot simply generalize to shop search problem. Two-stage methods cannot support a direct end-to-end optimization, which becomes the bottleneck of their expressive ability in specific CTR prediction task. Also, GIN doesn't take into account long-tail issue that is a serious phenomenon existing in shop search.

As a special scenario, shop search possesses several distinct characteristics: 1) user behaviors are quite sparser than item search, which is supported by the fact that daily page views of shop search only account for 1% of item search approximately; 2) complex heterogeneous relationships are involved in the context of shop search and the scale of the resultant heterogeneous graph is pretty

[1]http://taobao.com/

† Chenliang Li is the corresponding author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412087

large; 3) behavior sparsity gives rise to long-tail problem, most of the exposures are popular shops or queries of high frequency.

To address the above challenges, we propose a gated heterogeneous graph representation learning framework (GHL) for shop search. Firstly, to alleviate the behavior sparsity problem, we propose to transfer knowledge from item search. The item transaction history (*i.e.,* items purchased under a query and shops they belong to) are fully leveraged to enable better query understanding and retrieve more relevant shops. Incorporating knowledge of item search also enriches semantic information of long-tail queries and shops. Then to mine heterogeneous neighbor relationships encoded in the interaction graph, we exploit both first-order and second-order proximity between queries and shops. Weighted neighbor sampling strategy is employed by considering both scalability and efficiency. Moreover, we design a novel gated attentive neighbor aggregation scheme to assure that different types of entities (*i.e.,* long-tail ones and hot ones) can benefit from the heterogeneous graph automatically. Finally, the graph-based representations of queries and shops are fed into a two-tower architecture for relevance estimation and the whole framework is jointly trained in an end-to-end fashion.

The main contributions of this work can be summarized as follows:

(1) We propose to transfer knowledge from item search to relieve behavior sparsity problem existing in shop search.
(2) We utilize both first-order and second-order proximity of queries and shops encoded in the heterogeneous interaction graph and a novel gated attentive neighbor aggregation scheme is applied to enhance representation learning.
(3) Offline evaluation and online A/B tests verify the effectiveness of our proposed model.

2 THE PROPOSED APPROACH

In this section, we present the proposed method in detail, as shown in Figure 1. The overall architecture is a two-tower framework which is implemented as several MLPs with *LeakyReLU* as activation function. The most crucial part is how to build inputs for two-tower, *i.e.,* graph-based representations of queries and shops. The outputs of two-tower are fed into some relevance estimation function (*e.g.,* vector inner product) to get the final CTR prediction score.

2.1 Neighbor Construction

The underlying graph structure determines how much information can be assimilated from neighbors in order to strengthen the representation of node itself. Next we describe how we construct heterogeneous neighbors using both first-order and second-order proximity between queries and shops.

First-order proximity. We make use of click-through data from shop search directly. That is, the shops clicked under a query construct the query's first-order heterogeneous shop neighbors and vice versa. Due to the behavior sparsity issue, this first-order proximity derived from click-through data is very sparse. Therefore, we need to leverage second-order query-shop relationships through transferring knowledge from item search.

Second-order proximity. Item search contains more abundant information since it is the main search service provided by most e-commerce platforms. Here, for an item purchased under a query and provided by a shop, the corresponding query and shop are formulated as each other's second-order heterogeneous neighbor, respectively. Note that we did not utilize item click-through data since it might introduce much noise and purchase is a more reliable signal.

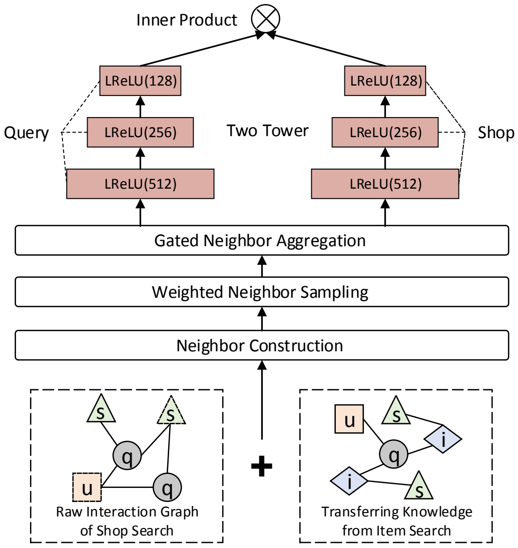

Inner Product
LReLU(128)
LReLU(128)
Query
LReLU(256)
Two Tower
LReLU(256)
Shop
LReLU(512)
LReLU(512)
Gated Neighbor Aggregation
Weighted Neighbor Sampling
Neighbor Construction
s
s
q
u
q
Raw Interaction Graph of Shop Search
u
s
q
i
i
s
Transferring Knowledge from Item Search

Figure 1: Framework of proposed GHL.

Weights assignment. After constructing heterogeneous neighbors, we further assign weight to each edge to indicate the relatedness between the query-shop pair. Combining information from both shop search and item search, we define the weight as the sum of numbers of the associated clicks and purchases.

2.2 Gated Attentive Neighbor Aggregation

In real-world e-commerce search, the size of heterogeneous neighbors of a query or a shop may vary significantly. To make computation more efficient, we adopt a **weighted neighbor sampling** strategy during training. The weights are calculated as stated in Section 2.1 and normalized over the heterogeneous neighbors of each node. Note that if some node has no neighbor, we instead use itself to fill the sampled set.

Next we describe how we aggregate information of sampled neighbors to enhance nodes (*i.e.,* queries and shops) representations. Take a piece of query q as example, let $\mathbf{h}_q$ be the embedding vector of q. Suppose we sample x heterogeneous neighbors (*i.e.,* shops) for q, the sampled set is denoted as $\mathcal{N}(q)$ and we have $|\mathcal{N}(q)| = x$. The neighbor aggregation is formulated as:

$$\mathbf{h}_{\mathcal{N}(q)} = AGGREGATE(\{\mathbf{h}_v, \forall v \in \mathcal{N}(q)) \quad (1)$$

where *AGGREGATE* is the aggregation function, such as averaging or max-pooling operation, $\mathbf{h}_{\mathcal{N}(q)}$ is the aggregated neighborhood representation of query q.

Here we propose to implement aggregation function with attention mechanism:

$$\mathbf{h}_{\mathcal{N}(q)} = \sum_{v \in \mathcal{N}(q)} \alpha_{q,v} \mathbf{h}_v \qquad (2)$$

where $\alpha_{q,v}$ is the attention weight, which is calculated with the guidance of query representation $\mathbf{h}_q$:

$$\alpha_{q,v} = \frac{exp(LeakyReLU(\mathbf{a}_q^\top [\mathbf{h}_q \| \mathbf{h}_v]))}{\sum_{k \in \mathcal{N}(q)} exp(LeakyReLU(\mathbf{a}_q^\top [\mathbf{h}_q \| \mathbf{h}_k]))} \qquad (3)$$

where $\|$ denotes vector concatenation, $\mathbf{a}_q$ is attention parameter for query, and *LeakyReLU* is the activation function.

We can simply take $\mathbf{h}_{\mathcal{N}(q)}$ as final representation of query q to establish a shop-to-shop framework. However, we need to discriminate the importance of aggregated neighborhood representation for different types of queries. For hot queries, the embedding vectors appear to be more critical since they are associated with too many neighbors that may hold massive noise. For long-tail queries, the heterogeneous neighbors usually are more decisive as we do not have sufficient prior knowledge of themselves.

Hence, inspired by GRU [1], we design a novel gated aggregation scheme to fuse the initial embedding vectors and corresponding neighborhood representations:

$$\mathbf{z} = \sigma(\mathbf{W}_z \cdot [\mathbf{h}_q \| \mathbf{h}_{\mathcal{N}(q)} \| \mathbf{h}_q^{pv}] + \mathbf{b}_z) \qquad (4)$$

$$\mathbf{r} = \sigma(\mathbf{W}_r \cdot [\mathbf{h}_q \| \mathbf{h}_{\mathcal{N}(q)} \| \mathbf{h}_q^{pv}] + \mathbf{b}_r) \qquad (5)$$

$$\hat{\mathbf{h}}_q = LeakyReLU(\mathbf{W} \cdot [(\mathbf{r} * \mathbf{h}_q) \| \mathbf{h}_{\mathcal{N}(q)}] + \mathbf{b}) \qquad (6)$$

$$\widetilde{\mathbf{h}}_q = (1 - \mathbf{z}) * \mathbf{h}_q + \mathbf{z} * \hat{\mathbf{h}}_q \qquad (7)$$

where $\mathbf{z}$ and $\mathbf{r}$ stand for update gate and reset gate respectively. In Eq. (4)-(5), $\mathbf{h}_q^{pv}$ represents **query pv (page view) features** which are used for distinguishing hot queries and long-tail queries. In practice, we embed discretized query page views and concatenate the resultant embeddings over different granularities to form $\mathbf{h}_q^{pv}$. As will be shown in experiments, these additional features boost the model performance explicitly.

Analogously, we can derive the final representation of shop s, denoted as $\widetilde{\mathbf{h}}_s$. The gated attentive neighbor aggregation for shop is identical except that the network parameters are distinct from the query side.

2.3 End-to-End Joint Training

After obtaining $\widetilde{\mathbf{h}}_q$ and $\widetilde{\mathbf{h}}_s$, we feed them into two-tower architecture to get the predicted click-through rate p_{qs} as shown in Figure 1. The outputs of the last layer of two-tower are carried out via a vector inner product and the result is sent into *sigmoid* function to meet the binary constraint.

The objective function is defined as cross entropy loss:

$$\mathcal{L} = -\sum_{(q,s)} y_{qs} log(p_{qs}) + (1 - y_{qs}) log(1 - p_{qs}) \qquad (8)$$

where y_{qs} is the ground truth label (*i.e.,* clicked or not clicked) of pair $< q, s >$.

In the training process, we leverage mini-batch training technique and the details of end-to-end joint training are presented in Algorithm 1.

Algorithm 1: Mini-batch Implementation of GHL

```
Initialize model parameters $\Theta$ randomly;
for iteration in 1, 2, ... do
    Pick a mini-batch of $B$ query-shop pairs $< q, s >$ and the
      corresponding pv features and labels $y_{qs}$;
    Sample by weight $(B \times 2x)$ heterogeneous neighbors
      $\mathcal{N}(q)$, $\mathcal{N}(s)$;
    Forward propagation to get $p_{qs}$ according to Eq. (2)-(7);
    Compute gradients $\nabla\mathcal{L}(\Theta)$ according to Eq. (8);
    Update model: $\Theta = \Theta - \epsilon\nabla\mathcal{L}(\Theta)$;
end
```

3 EXPERIMENTS

3.1 Experimental Setup

Dataset. We evaluate our proposed model based on a large-scale real-world dataset collected from Taobao users' search log with a period of 10 days. The dataset is comprised of 20 million queries, 7 million shops and 3.3 billion interactions, leading to data sparsity rate of 10^{-5}. The graph data contains 30 million nodes and 200 million edges, of which 70 percent are transferred from item search. We choose the first nine days as training set and leave tenth day for testing, which is called *Normal* test set. Moreover, we filter out a *Long Tail* test set where the queries or shops only appear once in training set.

Competitors. We conduct experiments with several competitors widely used in industrial community. (1) **Base**: the baseline two-tower model for large-scale search matching task, where we feed into two-tower network the query and shop initial embedding vectors and keep numbers of neurons of each layer as same as the proposed GHL for fair comparison. (2) **PinSage** [8]: the state-of-the-art GNN-based web-scale recommender system with GraphSage [3] as the backbone model. (3) **GAT** [6]: the self attention based GNN model. (4) **GHL**: our proposed method. Besides, we also report the performance of two variants of our model: **GHL-avg** replaces attentive aggregator (*i.e.,* Eq. (2)) with average pooling and **GHL-pv** removes the pv (page view) features during gated fusion.

Evaluation Metrics. We adopt three commonly used performance metrics for offline evaluation: *Area Under the receiver operating characteristic Curve* (*AUC*), *Group AUC* (*GAUC*), *Hit Ratio* (*HR@1*). *GAUC* is different from *AUC* as it measures the discrepancy between predictions and labels under each query and sum over all queries using the frequency as weight. Higher metric values demonstrate better search matching performance. Note that in our scenario a 0.001 increment means significant improvement.

3.2 Offline Evaluation

The experimental results of different methods on two test datasets (*i.e., Normal, Long Tail*) are presented in Table 1. As can be seen, GHL significantly outperforms all compared algorithms under two

Table 1: Overall performance of GHL and competitors.

Model	Normal			Long Tail		
	AUC	GAUC	HR@1	AUC	GAUC	HR@1
Base	0.8650	0.8528	0.7631	0.7195	0.6491	0.8794
PinSage	0.8657	0.8522	0.7637	0.7232	0.6534	0.8813
GAT	0.8593	0.8456	0.7519	0.7146	0.6448	0.8772
GHL-avg	0.8682	0.8545	0.7649	0.7382	0.6678	0.8883
GHL-pv	0.8669	0.8534	0.7640	0.7325	0.6613	0.8827
GHL	**0.8699**	**0.8560**	**0.7661**	**0.7416**	**0.6709**	**0.8926**

scenarios, especially on *Long Tail* test set. Specifically, the relative performance gains of *AUC*, *GAUC*, *HR@1* over the best baseline are 0.49%, 0.38%, 0.31%; 2.54%, 2.68%, 1.28% for *Normal* and *Long Tail* test set respectively. PinSage and GAT perform worse than GHL because they cannot fully tap the potential of heterogeneous relationships between queries and shops, and they do not take long-tail phenomenon into consideration. Moreover, GHL-avg attains inferior results to attentive aggregator and the decline of GHL-pv proves the contribution of incorporating pv features.

Neighbor count. We now analyze the effects of number of sampling neighbors. We vary neighbor count from 3 to 10 while keeping the other parameters fixed. The *AUC* values on both *Normal* and *Long Tail* datasets are depicted in Figure 2. We observe that the optimal results are obtained when neighbor count is 6 and either sampling more or less neighbors will lead to performance degradation. It is reasonable since a smaller neighbor count cannot provide enough information while a larger one may introduce too much noise.

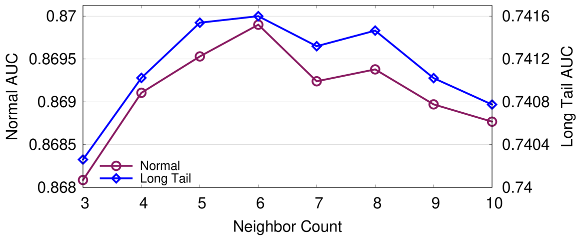
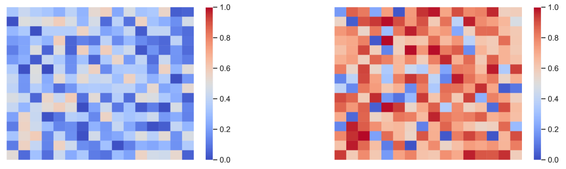


Figure 2: Comparison of different neighbor counts.

Case study. We further conduct some queries case study to visualize the update gate values (*i.e.,* $\mathbf{z}$ in Eq. (4)) in Figure 3. Figure 3(a) is a case of a hot query (*perfume*) whose representation relies more on its initial embedding vector while Figure 3(b) is another case where the long-tail query (*scarifier*) benefits more from heterogeneous neighbors. This verifies the correctness of gated fusion strategy in order to discriminate different types of queries.

3.3 Online A/B Test

We design an online A/B test (*i.e.,* bucket test) to further evaluate the performance of GHL. Table 2 shows online CTR improvements over Base model during 3 consecutive days. The long-tail shops are chosen on the basis of less than 10 exposures in training set. The results demonstrate the effectiveness of GHL on the shop search CTR prediction task, especially in solving long-tail problem.



(a) Hot query

(b) Long-tail query

Figure 3: Visualization of update gate values, which are reformed as matrices to ease display.

Table 2: Online CTR improvements of A/B test.

Time	T	T+1	T+2
whole bucket	2.67%	0.52%	3.70%
long-tail shops	19.71%	18.41%	24.34%

4 CONCLUSION

In this paper, we introduce a gated heterogeneous graph representation learning framework for shop search in e-commerce. A serious problem, *i.e.,* behavior sparsity, is alleviated through transferring knowledge from item search. Both first-order and second-order proximity of queries and shops contained in the heterogeneous graph are mined to enhance representations. A novel gated attentive neighbor aggregation strategy is leveraged such that different types of entities can benefit from heterogeneous relationships in an automatic manner. Experiments on offline and online real-world dataset demonstrate the effectiveness of our proposed model, especially for long-tail queries and shops.

REFERENCES

[1] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *EMNLP*. 1724–1734.

[2] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. 855–864.

[3] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1024–1034.

[4] Feng Li, Zhenrui Chen, Pengjie Wang, Yi Ren, Di Zhang, and Xiaoyu Zhu. 2019. Graph Intention Network for Click-through Rate Prediction in Sponsored Search. In *SIGIR*. 961–964.

[5] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*. 701–710.

[6] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[7] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *KDD*. 839–848.

[8] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*. 974–983.

[9] Yuan Zhang, Dong Wang, and Yan Zhang. 2019. Neural IR Meets Graph Embedding: A Ranking Model for Product Search. In *WWW*. 2390–2400.