NEIGHBOR-AUGMENTED TRANSFORMER-BASED EMBEDDING FOR RETRIEVAL

Jihai Zhang[1], Fangquan Lin[1], Wei Jiang[1], Cheng Yang[1], Gaoge Liu[2]

[1]Alibaba Group, China
jihai.zjh@alibaba-inc.com
[2] Columbia University, USA

ABSTRACT

With rapid evolution of e-commerce, it is essential but challenging to quickly provide a recommending service for users. The recommender system can be divided into two stages: retrieval and ranking. However, most recent academic research has focused on the second stage for datasets with limited size, while the role of retrieval is heavily underestimated. Generally, graph-based or sequential models are used to generate item embedding for the retrieval task. However, the graph-based methods suffer from over-smoothing, while sequential models are largely influenced by data sparseness. To alleviate these issues, we propose NATM—a novel embedding-based method in large-scale learning incorporating both graph-based and sequential information. NATM consists of two key components: *i)* neighbor augmented graph construction with user behaviors to enhance item embedding and mitigate data sparseness, followed by *ii)* transformer-based representation network, targeting on minimizing NCE loss. The competitive performance of the proposed method is demonstrated through comprehensive experiments, including a benchmark study on MovieLens dataset and a real-world e-commerce scenario in Alibaba Group.

Index Terms— embedding, neighbor graph, transformer

1. INTRODUCTION

In general, the recommender system can be divided into two stages. Firstly, the retrieval stage aims to retrieve potentially interesting items to users from a massive item library, mainly based on user behavior. Then the ranking stage follows, which has the ability to incorporate more side information and to develop complex models to accurately make personalized recommendation. Usually in academic research, since the size of dataset is limited, an end-to-end task is performed to directly rank all the items. However, in industrial applications, it is unrealistic to rate and rank millions of items one by one when a user browses online. As a result, the retrieval stage is particularly important and its efficiency affects the performance of the downstream ranking task.

As one of the earliest recommendation algorithms, Collaborative Filtering (CF) [1] analyzes the relationship among items to measure their similarity. However, modeling the static user-item interactions results in a homogeneous recommendation, and the scalability issue is challenging since there are millions of items and users in the real-world purchasing scenarios. To more effectively capture a user's potential interest, many recommendation algorithms focus on learning item embedding [2, 3, 4], *i.e.,* mapping items to a low-dimensional vector space, and capturing the implicit relationship. Recently, deep learning has revolutionized the recommender system dramatically. For instance, Covington et al. [5] propose Deep Match for YouTube recommendation by embedding the user behavior sequence and the corresponding user profile. Notably, recent deep learning techniques have mainly advanced in the graph-based methods [6] and the sequential recommendation [7, 8] to capture the structured dependency in user behavior. The item embedding obtained in the above methods can be used to measure the item-to-item distance for the retrieval task[9]. Still, the graph-based methods often encounter the problem of over-smoothing [10] and scalability [11] of model parameters with the growth of the network depth. Similarly, although the recent sequential recommender can be equipped with various user modules such as long-term preferences [8, 12] and multi-interest [13] for diverse predictions, it is strongly affected by the issue of data sparseness [14]. Indeed, representations learned for the low-frequency items are noisily, and the inaccurate embedding can influence the robustness of the entire network [15].

To address these issues, we propose a novel Neighbor-Augmented Transformer-Based Embedding Model (short for NATM) to ameliorate item-to-item retrieval task. NATM adopts the transformer to encode an item, together with its co-occurred item(s) as neighbors to augment the representation. Moreover, positive and negative examples are generated with contrastive learning for model training. To the best of our knowledge, this is the first work that the neighbors information is combined with sequential modelling in the framework of item-to-item retrieval. Furthermore, offline and online experiment results demonstrate its effectiveness over the state-of-the-art graph-based and sequential methods on both public datasets and a real-world scenario in Alibaba.

[1]Equal contribution and Jihai Zhang is the corresponding author.

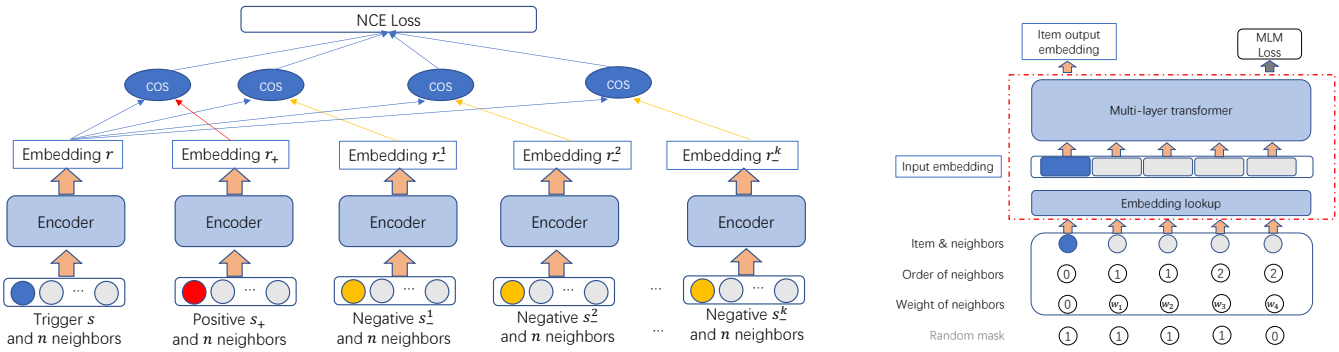

Fig. 1. Model architecture for NATM (left plot). The twined-structured model consists of the main item and its neighbors as input, the encoder to obtain the embedding vector. Specifically, the encoder (right plot) is a transformer, with the order and weight of neighbors as position embedding. The training objective is to minimize the weighted sum of NCE and MLM loss.

2. THE MODEL

2.1. Problem Statement

We focus on the task of item-to-item retrieval by taking online shopping as an illustrative example. When a user is browsing the main page of an item A, he has potential to click a related item B. In our framework, the item A is called as the trigger and B the target, which belongs to a set of candidate items. Given a trigger item, the objective is to retrieve the target item from a massive library of candidates.

More formally, the input consists of user sequence of length N as $S = (s_1, s_2, ..., s_N)$, where the element $s_i \in \mathcal{I}$ indicates the i-th item that the user visited in order. Given a set of user sequences, samples can be reconstructed in an item-based way to summarize the global user behavior as the form of (s, s_c, y), where s denotes the trigger item and s_c denotes the candidate one. The candidate item can either interact with the trigger item or not depending on the value of the label $y \in \{1, 0\}$. The problem then turns to learn the item representation given the trigger-candidate pairs.

2.2. Model Architecture

The proposed model is based on two-tower structure, which was first derived as a deep structured semantic model [16] for the query-document matching in web search. The query and document are projected to low-dimensional semantic vectors based on the massive click-exposure logs, and the distance between two vectors is calculated by the cosine similarity. Usually, the models of query and document are separated into two sub-networks as a twined structure.

In previous retrieval approach [8], sub-networks are built for user and item respectively. Alternatively, the NATM extends the two-tower structure for the trigger item and the candidate one, in the context of item-to-item retrieval task. Moreover, to tackle the scalability issue, the framework of noise contrastive estimation (NCE)-based learning [17] is adopted. By sampling k negative candidates, the trigger-candidate pairs (s, s_c, y) are transformed to $(s, s_+, s_-^1, \cdots, s_-^k)$, where s_+ denotes the positive example and $(s_-^1, \cdots, s_-^k)$ denotes a set of k negative examples. Then the model can be trained by contrasting positive and negative examples and learning the discriminative item representation as detailed in Section 2.4.

Figure 1 demonstrates the model architecture of NATM. This two-tower model has two key components to obtain an enhanced item representation: one is the encoder design and the other is the construction of positive and negative samples.

Encoder. Multimodal fusion of input is considered to capture the rich information in item features. Furthermore, the input of the encoder also contains the features of the item's neighbors, as detailed in Section 2.3. In addition, as illustrated in Figure 1 (right), multi-layer transformer is applied to encode the structured input of items and their neighbors. Indeed, this structured input can be regarded as a "quasi" sequence, distinguished from the original user sequence.

Sampling. Positive examples can be obtained by retrieving two adjacent items in a user sequence. And negative examples can be generated by random sampling from all candidates.

2.3. Construction of Neighbor Graph

The classical deep retrieval model fails to represent the low-frequency items [15]. To tackle the sparsity issue, NATM enhances the item embedding by taking into account its relevant neighbors. The neighbor graph can be constructed as follows: the nodes represent the items, and the edges represent the frequency of co-occurrence of the corresponding two items (*i.e.,* interacted sequentially by a user). In addition, the edge of the neighbor graph is weighted according to the number of interactions summarized from the global user behavior. Figure 2 represents a subgraph of the whole neighbor graph. Then the two-tower model can be built for trigger and candidate items with their neighbors, as illustrated in Figure 1 (left).

Note that this way of graph construction deals with the noise in the user behavior sequence. Take online shopping

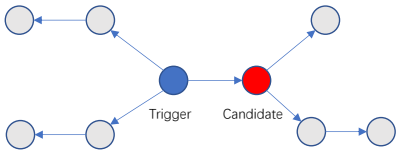

Fig. 2. Neighbor graph of items.

as an example: after viewing several mobile phones, the user may change his/her interest and click a T-shirt. In such cases, the pair of trigger-candidate items may have little relevance, but the regular modeling involves such noise. To handle this issue, in NATM, only when two items co-occur frequently, will they appear together as neighbored input. In addition, the transformer can also suppress the noise from high-order neighbors by taking smaller weights in self-attention[18].

2.4. Model Learning

MLM task. As illustrated in Figure 1 (right), for each subgraph as input, the multi-layer transformer performs item-level masked language model (MLM) task [19]. More precisely, we randomly mask a portion of items on the subgraph then predict the original ids of the masked items based on the surrounding context. Through the transformer layer, we obtain the item representation, with the augmented information provided by its neighbors. In addition, position embedding is introduced to distinguish the order of neighbors. Moreover, the number of interactions across the item and its neighbors is counted as the weight embedding.

NCE loss. After that, the transformer layer outputs the item embedding, which is then utilized to calculate NCE loss. We first define the relevance score between two items s_1 and s_2 as: $\mathcal{R}(s_1, s_2) = \text{cosine}(r_1, r_2) = \frac{r_1^T r_2}{\|r_1\|\|r_2\|}$, with the corresponding embedding vector r_1 and r_2.

Calculating the commonly used softmax cross-entropy is computationally expensive due to an enormous number of candidate items. Therefore, NCE transfers the objective into a binary classification by considering k negative examples:

$$p(y=0|s, s_c) = \frac{kq(s_c)}{\exp\left(\gamma \mathcal{R}(s, s_c)\right) + kq(s_c)},$$

$$p(y=1|s, s_c) = \frac{\exp\left(\gamma \mathcal{R}(s, s_c)\right)}{\exp\left(\gamma \mathcal{R}(s, s_c)\right) + kq(s_c)},$$

where $y = 0$ indicates that s_c is a negative example of s while $y = 1$ for the positive case; γ is the smoothing factor; and $q(\cdot)$ is the noise distribution. The NCE loss is defined as follows:

$$\mathcal{L}_{\text{NCE}} = \sum_{(s, s_c)} \left(\log p(y=1|s, s_c) + k\mathbb{E}_{\bar{s}_c \sim q} \log p(y=0|s, s_c)\right).$$

During the training stage, the set of parameters θ is estimated by minimizing the sum of NCE loss and MLM loss. Then during the testing stage, given a new trigger item, top related items can be retrieved by efficient similarity search [20] through embedding vectors of massive candidates.

3. EXPERIMENTS

3.1. Experiment settings

Datasets. To evaluate the proposed method, we conduct extensive experiments on two datasets: MovieLens[1] and a real e-commerce dataset on Alibaba Taobao. MovieLens contains movie ratings for each logging user and we convert all numeric ratings of a user to a sequence of positive implicit feedback. Similarly, the feedback logs on Taobao contain the user click sequences. The objective of the prediction task is to retrieve the last element in each sequence given the precedent ones. The statistics of the processed offline datasets are summarized in Table 1.

Table 1. Statistics of the offline experiment datasets.

Dataset	# users	# items	# actions	Avg.length
MovieLens-1m	6k	3.4k	1.0m	164
Taobao.com	20m	100m	0.6b	30

Metrics. The precision at K(Pre@K) represents the proportion of cases which has the correct prediction among the top K items. Pre@K is an offline metric. Click Through Rate (CTR) is the number of clicks divided by item exposure times. Gross Merchandise Volume (GMV) is the gross revenue. CTR and GMV are online metrics.

Comparison Methods. First, we consider **POP**—the simplest baseline that always recommends the most popular items. Three representative and widely adopted graph-based approaches are selected: **GCN** [21], **GAT** [22] and **EGES** [23]. Specifically, GAT adopts the attention mechanism to learn the relative weights between two connected nodes and EGES learns node's representation based on the neighborhood network with a random walk. Furthermore, sequential models are also taken into account, including **SDM** [8]—a sequential deep matching model and **Bert4Rec** [7]—a transformer-based model inspired by Bert. Additionally for online setting, we compare with **ESM2** [24]—a multi-task model currently deployed on Taobao.

3.2. Results

Offline results. As shown in Table 2, the proposed method outperforms others in terms of all the retrieval metrics, which demonstrates the effectiveness of our proposed model. Note that most baseline methods can be implemented for MovieLens. In contrast, for the real-world large-scale e-commerce Taobao dataset, additional efforts have been required to optimize the computational performance, including distributed data storage, parallel training and NCCL communicator. Unfortunately, graph-based methods do not demonstrate the ability to be reasonably trained in the large-scale learning. Therefore, the evaluation of GCN and GAT is only provided for MovieLens. Meanwhile, we note that MovieLens contains

[1]https://grouplens.org/datasets/movielens/

only the user rating without any side information such as user features. However, SDM requires user features as a query to activate the attention mechanism, as its twined structure is built on both user and item sides. For this concern, SDM can be evaluated only on the e-commerce Taobao dataset, which provides the necessary user information.

Table 2. Method comparison on retrieval metrics. Bold scores are the best, while underlined scores are the sub-optimal. The symbol $*$ indicates statistical significance of improvement compared to the best baseline measured by t-test at p-value of 0.05.

Methods	MovieLens-1m			Taobao.com		
	Pre@1	Pre@5	Pre@50	Pre@1	Pre@5	Pre@100
POP	0.0030	0.0078	0.1035	0.0858	0.2872	0.7993
GCN	0.0066	0.0247	0.1457	-	-	-
GAT	0.0091	0.0329	0.1909	-	-	-
EGES	0.0062	0.0280	0.2079	0.1080	0.3910	0.8685
SDM	-	-	-	0.1262	0.3955	0.8733
Bert4Rec	0.0067	0.0330	0.1748	0.1159	0.3633	0.8526
NATM	**0.0127***	**0.0449***	**0.2402***	**0.1638***	**0.4700***	**0.9026***
Improv.	39.56%	36.06%	15.54%	29.79%	18.84%	3.36%

Online results. The baseline ranking model on Taobao online environment is Entire Space Multi-Task Model (ESM2) [24]. We compare the performance of ESM2 with and without the item embedding of NATM through online A/B test. Each deployed method has involved the same number of users (more than one million). As shown in Figure 3, NATM contributes to 3% CTR and 2.5% GMV lifting on average in one week. These results indicate the significant commercial value of the proposed method for the e-commerce platform.

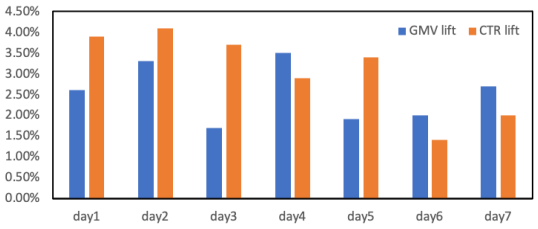

Fig. 3. Lift in CTR and GMV of NATM pretrain compared to ESM2 baseline in online A/B test .

3.3. Ablation study

We perform ablation experiments over a number of key components in NATM to better understand their impact on retrieval. Due to space constraints, we only show the representative results on MovieLens-1m.

Layer number of self-attention. As shown in existing transformer-based models such as Bert4Rec, the number of layers L of self-attention is one of the most important parameters. According to Table 3, we observe that a more complex hidden structure may not lead to better performance. Indeed, the volume of model parameters expands rapidly as the number of layers increases, which challenges the capacity of generalization. In this dataset, the best results are obtained when the number of layers equals 4. Here the 0 layer represents that the model contains only multilayer perception and feedforward neural network. The effectiveness of the transformer-based encoder is demonstrated, for example, Pre@50 of the model with 4-layer transformer is 27% higher than that without the transformer.

Table 3. Effect of the layer number of self-attention.

Metrics	0	1	2	3	4	8
Pre@50	0.138	0.172	0.171	0.170	**0.175**	0.146

Number of neighbors. As shown in Table 4, the optimal number of neighbors is 9 empirically, which indicates a balance of information and noise the neighbor augmentation brings. We also observe that incorporating neighbor augmentation (number of neighbors > 0) can achieve better results than excluding it (number of neighbors $= 0$). Specifically, 9 neighbors provide a 13% gain in Pre@50.

Table 4. Effect of neighborhood number.

Metrics	0	1	5	8	9	10
Pre@50	0.158	0.170	0.177	0.175	**0.178**	0.177

Number of negative samples As introduced in Section 2.4, negative samples are generated to build the model and obtain the NCE loss. According to Table 5, when the number of negative samples k increases, the retrieval performance also augments and finally tends to converge to a stationary value around 0.208 in Pre@50.

Table 5. Effect of negative samples.

Metrics	1	3	5	7	9	11
Pre@50	0.129	0.174	0.194	0.200	**0.208**	0.207

Note that the ablation study in one component has been conducted with all other hyper-parameters equal. In the experiment of method comparison as introduced in Section 3.2, by testing all the possible combinations of hyper-parameters, consequently, the optimal results are summarized in Table 2.

4. CONCLUSION

In this paper, we propose a novel embedding-based deep retrieval model — NATM for item-to-item retrieval task, to retrieve top candidate items from a large library of items efficiently. Besides, the proposed method can be overlaid on the downstream ranking models to achieve a better performance. The main contribution comes from two parts, including neighbor augmented graph construction, followed by the transformer-based learning stage in the framework of contrastive learning. Finally, we empirically demonstrated the effectiveness of our approach according to the comprehensive offline and online experiments on the public dataset MovieLens and in a real-world e-commerce scenario in Alibaba Group as well.

5. REFERENCES

[1] Greg Linden, Brent Smith, and Jeremy York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[2] Oren Barkan and Noam Koenigstein, "Item2vec: neural item embedding for collaborative filtering," in *MLSP*. IEEE, 2016, pp. 1–6.

[3] Fangquan Lin, Wei Jiang, Jihai Zhang, and Cheng Yang, "Dynamic popularity-aware contrastive learning for recommendation," in *ACML*. PMLR, 2021, pp. 964–968.

[4] Wei Jiang, Fangquan Lin, Jihai Zhang, Cheng Yang, Hanwei Zhang, and Ziqiang Cui, "Dynamic sequential recommendation: Decoupling user intent from temporal context," in *ICDMW*. IEEE, 2021, pp. 18–26.

[5] Paul Covington, Jay Adams, and Emre Sargin, "Deep neural networks for youtube recommendations," in *RecSys*, 2016, pp. 191–198.

[6] Aditya Grover and Jure Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*, 2016, pp. 855–864.

[7] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *CIKM*, 2019, pp. 1441–1450.

[8] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng, "Sdm: Sequential deep matching model for online large-scale recommender system," in *CIKM*, 2019, pp. 2635–2643.

[9] Hui Wu, Min Wang, Wengang Zhou, Yang Hu, and Houqiang Li, "Learning token-based representation for image retrieval," *arXiv preprint arXiv:2112.06159*, 2021.

[10] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," *AAAI*, 2020.

[11] Qiaoyu Tan, Ninghao Liu, Xing Zhao, Hongxia Yang, Jingren Zhou, and Xia Hu, "Learning to hash with graph neural networks for recommender systems," in *WWW*, 2020, pp. 1988–1998.

[12] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai, "Practice on long sequential user behavior modeling for click-through rate prediction," in *KDD*, 2019, pp. 2671–2679.

[13] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee, "Multi-interest network with dynamic routing for recommendation at tmall," in *CIKM*, 2019, pp. 2615–2623.

[14] Guorui Zhou, Kailun Wu, Weijie Bian, Xiaoqiang Zhu, and Kun Gai, "Res-embedding for deep learning based click-through rate prediction modeling," in *DLP-KDD*, 2019.

[15] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen, "Challenging the long tail recommendation," *Proc. VLDB Endow.*, vol. 5, no. 9, pp. 896–907, May 2012.

[16] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck, "Learning deep structured semantic models for web search using clickthrough data," in *CIKM*, 2013, pp. 2333–2338.

[17] Michael U Gutmann and Aapo Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics.," *Journal of Machine Learning Research*, vol. 13, no. 2, 2012.

[18] Jianbo Ouyang, Hui Wu, Min Wang, Wengang Zhou, and Houqiang Li, "Contextual similarity aggregation with self-attention for visual re-ranking," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[20] Jeff Johnson, Matthijs Douze, and Hervé Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2021.

[21] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[22] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio, "Graph attention networks," in *ICLR*, 2018.

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

[24] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang, "Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction," in *SIGIR*, 2020, pp. 2377–2386.