A Dual Augmented Two-tower Model for Online Large-scale Recommendation

Yantao Yu, Weipeng Wang, Zhoutian Feng, Daiyue Xue
Meituan
Beijing, China
{yuyantao,wangweipeng02,fengzhoutian,xuedaiyue}@meituan.com

ABSTRACT

Many modern recommender systems have a very large corpus, and a common industrial recipe for handling large-scale retrieval is to learn query and item representations from their content features with the *two-tower* model. However, the model suffers from lack of information interaction between the two towers. Besides, imbalanced category data also hinders the model performance. In this paper, we propose a novel model named *Dual Augmented Two-tower Model* (DAT), which integrates a novel Adaptive-Mimic Mechanism (AMM) and a Category Alignment Loss (CAL). Our AMM customizes an augmented vector for each query and item to mitigate the lack of information interaction. Moreover, our CAL can further improve performance by aligning item representation of uneven categories. Offline experiments on large-scale datasets are conducted to show the superior performance of DAT. Moreover, online A/B testings confirm that DAT can lead to improved recommendation quality for industrial applications.

CCS CONCEPTS

• **Information systems → Recommender systems**; *Information retrieval.*

KEYWORDS

Recommender Systems, Information Retrieval, Neural Networks

ACM Reference Format:
Yantao Yu, Weipeng Wang, Zhoutian Feng, Daiyue Xue. 2021. A Dual Augmented Two-tower Model for Online Large-scale Recommendation. In *Proceedings of DLP-KDD 2021.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recommender systems are indispensable in filtering out items that users are interested in from huge amounts of items. One of the most crucial challenges in large-scale recommendation is to score millions or billions of items in real-time accurately. A common practice is to design the recommender as a two-phase architecture where a retrieval model first retrieves a small fraction of related items given user's query from a large corpus, and a ranking model ranks the retrieved items based on clicks or user-ratings [2]. Obviously, the quality of candidates retrieved in the retrieval stage plays a critical role in the whole system. In this work, we focus on the retrieval problem with an aim to improve the retrieval system's performance for personalized recommendation with millions of queries and items.

A scalable retrieval model usually learns query and item representations first and then uses a cosine similarity between the query and item representations to obtain recommendations tailored for the query. However, in industrial-scale applications, the corpus of items can be enormously large and the training data collected from users' feedback is likely to be very sparse for most queries and items, which may lead to inaccurate model predictions for long-tail users and items. To tackle the above challenges, the two-tower model [4], with towers referring to encoders based on deep neural networks (DNNs), are usually applied. Despite great promise, there are still some problems in two-tower model. As the item representation of the item tower must be pre-computed separately for online retrieval service, the forward computation of the item tower must be independent with the query tower, and so the model lacks information interaction between the two towers, which may inevitably hinder the model's performance. In real-world applications, the input of one tower has a positive interaction with the input of the other for each click. For example, suppose that item A is clicked in queries $\{B, C, D\}$. Apparently, the inputs for the query tower are $\{B, C, D\}$, which interact with the input A of the item tower. Consequently, the information contained in the query tower can be leveraged to augment the item representation of the item tower, and vice versa. Furthermore, the categories of items are diverse (e.g., food, hotels, movies, etc.) and the amount of items in each category is imbalanced severely. Thus the items in an individual category may account for the majority. Consequently, model performs much worse on the categories with relatively smaller amount of items.

In this paper, to tackle the above issues, we present a novel retrieval model for large-scale recommendation named *Dual Augmented Two-tower Model* (DAT).Specifically, we design an Adaptive-Mimic Mechanism to customize an augmented vector for each query and item as their content features. The augmented vector is updated according to the output representation vector of the other tower for each sample with positive labels. In this way, the augmented vector as the input feature of one tower carries valuable information of the other tower, which implicitly models the information interaction between the two towers. And we also introduce a Category Alignment Loss in the training phase to align the representation for items from different categories. Comprehensive experiments show that our DAT features have two major advantages: i). it provides deeper insights into the information interaction of two-tower models in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
DLP-KDD 2021, August 15, 2021, Singapore
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

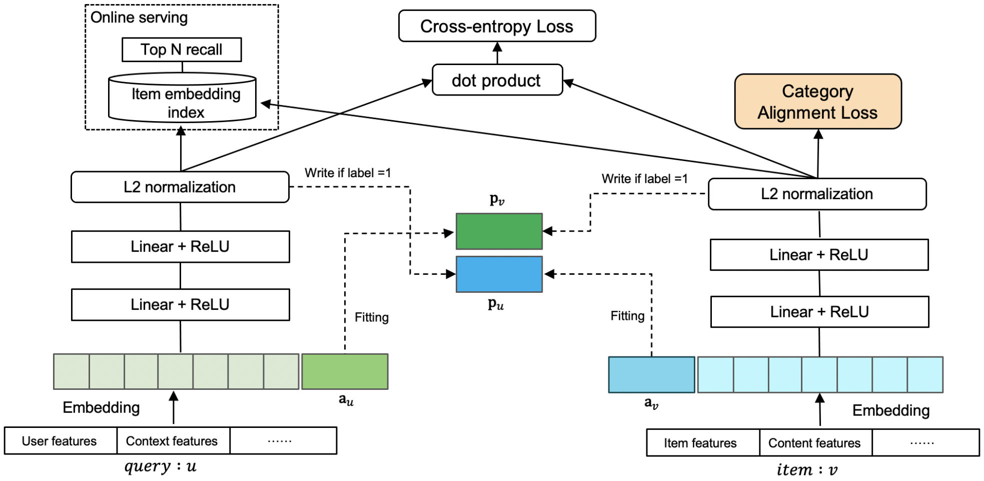

Figure 1: The network architecture of our proposed Dual Augmented Two-tower Model

the retrieval task; ii). it produces better item representations when the category distribution is extremely imbalanced.

2 MODEL ARCHITECTURE

2.1 Problem Statement

We consider a recommendation system with a query set $\{\mathbf{u}_i\}_{i=1}^{N}$ and an item set $\left\{\mathbf{v}_j\right\}_{j=1}^{M}$, where N is the number of users and M is the number of items. Here $\mathbf{u}_i$, $\mathbf{v}_j$ are the concatenations of various features (e.g., IDs and content features), which can be very high-dimensional due to the sparsity. The query-item feedback can be represented by a matrix $R \in \mathbb{R}^{N\times M}$, where $R_{ij} = 1$ if query i gives a positive feedback on item j, and $R_{ij} = 0$ otherwise. Our objective is to efficiently select possibly thousands of candidate items from the entire item corpus given a certain query.

2.2 Dual Augmented Two-tower Model

The framework of our proposed model is illustrated in Figure 1. The DAT model uses an augmented vector $\mathbf{a}_u(\mathbf{a}_v)$ to capture the information from the other tower, and regards the vector as the input feature of one tower. Additionally, the Category Alignment Loss transfers the knowledge learned in the category with the largest amount of data to other categories

2.2.1 Embedding Layer. Similar to two-tower models, each feature $\mathbf{f}_i \in \mathbb{R}$ (e.g., an item ID) in $\mathbf{u}_i$ and $\mathbf{v}_j$ goes through an embedding layer and is mapped to a low-dimensional dense vector $\mathbf{e}_i \in \mathbb{R}^K$, where K is the embedding dimension. Specifically, we define an embedding matrix $\mathbf{E} \in \mathbb{R}^{K\times D}$ where $\mathbf{E}$ is to be learned and D is the number of unique features, and the embedding vector $\mathbf{e}_i$ is the ith column of the embedding matrix $\mathbf{E}$.

2.2.2 Dual Augmented layer. For a certain query and candidate item, we create two corresponding augmented vectors $\mathbf{a}_u$ and $\mathbf{a}_v$ by their IDs, and concatenate them with feature embedding vectors to obtain the augmented input vectors $\mathbf{z}_u$, $\mathbf{z}_v$ of the two towers. For example, if query u has features "uid=253,city=SH,gender=male,..." and item v has features "iid=149,price=10,class=cate,...", we have:

$$
\begin{aligned}
\mathbf{z}_u &= [\mathbf{e}_{253} \,\|\mathbf{e}_{sh}\|\, \mathbf{e}_{male} \,\|...\|\, \mathbf{a}_u] \\
\mathbf{z}_v &= [\mathbf{e}_{149} \,\|\mathbf{e}_{p10}\|\, \mathbf{e}_{cate} \,\|...\|\, \mathbf{a}_v]
\end{aligned}
$$

where $||$ is the vector concatenation operation. The augmented input vectors $\mathbf{z}_u$ and $\mathbf{z}_v$ not only contain information about the current query and item, but also contain information about historical positive interactions through $\mathbf{a}_u$ and $\mathbf{a}_v$.

Next, we feed $\mathbf{z}_u$ and $\mathbf{z}_v$ into the two towers, which are composed of fully connected layers with the ReLU activation function, in order to achieve the information interaction between the two towers by $\mathbf{a}_u$ and $\mathbf{a}_v$ (augmented vectors). Next, the output of the fully connected layers goes through an L2 normalization layer to get augmented representations of query $\mathbf{p}_u$ and item $\mathbf{p}_v$. Formally, the definition of the two steps is as follows:

$$
\begin{aligned}
\mathbf{h}_1 &= ReLU\left(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1\right), \ldots \\
\mathbf{h}_L &= ReLU\left(\mathbf{W}_l\mathbf{h}_{L-1} + \mathbf{b}_l\right), \\
\mathbf{p} &= L2Norm\left(\mathbf{h}_L\right)
\end{aligned}
\tag{1}
$$

where $\mathbf{z}$ denotes $\mathbf{z}_u$ and $\mathbf{z}_v$, $\mathbf{p}$ denotes $\mathbf{p}_u$ and $\mathbf{p}_v$; $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weight matrix and bias vector for the l th layer, respectively; $\mathbf{p}_u$ and $\mathbf{p}_v$, the output vectors of the L2 normalization layer, represent the query embedding and item embedding, respectively. The two towers have the same structure but different parameters.

Furthermore, to estimate the augmented vectors $\mathbf{a}_u$ and $\mathbf{a}_v$, we design an Adaptive-Mimic Mechanism (AMM), which integrates a mimic loss and a stop gradient strategy. The mimic loss aims to use the augmented vector to fit all positive interactions in the other tower belonging to the corresponding query or item. We define mimic loss as the mean square error between the augmented vector and query/item embedding $\mathbf{p}_u$,$\mathbf{p}_v$ for each sample of which label equals to 1 :

$$
\begin{aligned}
loss_u &= \frac{1}{T}\sum_{(u,v,y)\in\mathcal{T}}\left[y\mathbf{a}_u + (1-y)\mathbf{p}_v - \mathbf{p}_v\right]^2 \\
loss_v &= \frac{1}{T}\sum_{(u,v,y)\in\mathcal{T}}\left[y\mathbf{a}_v + (1-y)\mathbf{p}_u - \mathbf{p}_u\right]^2
\end{aligned}
\tag{2}
$$

where T is the number of query-item pairs in the training dataset $\mathcal{T}$, and $y \in \{0,1\}$ is the label. We discuss the labeling process in the next sub-section. As can be seen, if the label $y = 1$, $\mathbf{a}_v$ and $\mathbf{a}_u$ approach the query embedding $\mathbf{p}_u$ and item embedding $\mathbf{p}_v$; if the label $y = 0$, the loss is is equal to 0. As shown in Figure 1, the augmented vectors are used in one tower and the query/item embeddings are generated from the other. That is to say, the augmented vectors $\mathbf{a}_u$ and $\mathbf{a}_v$ summarize the high-level information about what a query or an item possibly matches from the other tower. Since the mimic loss is to update $\mathbf{a}_u$ and $\mathbf{a}_v$, we should freeze the value of $\mathbf{p}_u$ and $\mathbf{p}_v$. To do so, the stop gradient strategy is applied to stop the gradient of $loss_u$ and $loss_v$ from flowing back into $\mathbf{p}_v$ and $\mathbf{p}_u$.

Once the two augmented vectors $\mathbf{a}_u$ and $\mathbf{a}_v$ are obtained, they are used to model the information interaction between the two towers by regarding them as the input feature of the two towers. Finally, the output of the model is the inner product of the query embedding and item embedding:

$$s(\mathbf{u}, \mathbf{v}) = \langle \mathbf{p}_u, \mathbf{p}_v \rangle$$

where $s(\mathbf{u}, \mathbf{v})$ denotes the score provided by our retrieval model.

2.2.3 Category Alignment. In the industrial scenario, the categories of items will be diverse (e.g., food, hotels, movies, etc.) and the number of items in each category is seriously uneven. With the imbalanced category data, the two-tower model performs differently for the different categories, and it performs much worse on the categories with relatively smaller amount of items. To tackle the problem, we propose a Category Alignment Loss (CAL) for training phase, which transfers the knowledge learned in the categories with a large amount of data to other categories. Specifically, for each batch, the item representation $\mathbf{p}_v$ of the category with the largest amount of data is taken to form the major category set: $S^{major} = \left\{\mathbf{p}_v^{major}\right\}$, and the $\mathbf{p}_v$ of other categories form their respective category sets:$S^2, S^3, S^4, ...$ We define the category alignment loss as the distance between the second-order statistics (covariances) of the major category and other categories features:

$$loss_{CA} = \sum_{i=2}^{n} \left\| \mathrm{C}(S^{major}) - \mathbf{C}(S^i) \right\|_F^2 \tag{3}$$

where $\|\cdot\|_F^2$ denotes the squared matrix Frobenius norm and n is the number of categories. $\mathbf{C}(\cdot)$ represents the covariance matrix.

2.3 Model Training

We treat the retrieval problem as a binary classification problem, and employ a random negative sampling framework. Specifically, for the query in each positive query-item pair (label = 1), we randomly sample S items from the item corpus to create S negative query-item pairs (label = 0) with this query, and add these $S + 1$ pairs to the training dataset. The cross-entropy loss for these pairs is as follows:

$$loss_p = -\frac{1}{T} \sum_{(u,v,y) \in \mathcal{T}} (y log \sigma(\langle \mathbf{p}_u, \mathbf{p}_v \rangle) + (1-y) \, log \, (1 - \sigma(\langle \mathbf{p}_u, \mathbf{p}_v \rangle)))$$
$$T = D \times (S+1) \tag{4}$$

where D is the number of positive feedback query-item pairs and T is the total number of train pairs. $\sigma(\cdot)$ denotes the sigmoid function.

The final loss function is formulated as:

$$loss = loss_p + \lambda_1 loss_u + \lambda_2 loss_v + \lambda_3 loss_{CA} \tag{5}$$

where $\lambda_1, \lambda_2, \lambda_3$ are tunable parameters.

Table 1: Statistics of Datasets.

Dataset	#users	#items	#interactions	#categories
Amazon Books	695,513	243,166	6,706,125	11
Meituan	82,347,274	3,561,498	1,182,652,197	9

3 EXPERIMENTS

In this section, we conducted extensive online and offline experiments to justify the rationality of DAT's design.

3.1 Datasets

All models were evaluated on two offline large-scale datasets: a large dataset sampled from the daily logs of online systems on Meituan[1] and a dataset from Amazon [3]. The Meituan dataset contains data within 11 consecutive days in which the data of the first 10 days was for training and the data of the $11th$ day was for testing and we combined all the items that appeared in the first 10 days as the item corpus. The Amazon Books is relatively smaller, and we only kept items that have been reviewed for least 5 times and users who have reviewed at least 5 items. We left the last item out for testing. The details of the two offline datasets are listed in Table 1.

3.2 Experimental Setup

The following methods which were widely applied to industrial were used in the comparison with the proposed DAT model:

- **WALS** [1] A matrix factorization algorithm for decomposing the interaction matrix into hidden factors of users and items.
- **YouTubeDNN** [2] A deep neural network based recommendation approach that fed vectors into a multi-layer feed forward neural network.
- **FM** [7] A model that accumulates the feature vectors of query and item and feeds them into the FM layer.
- **Two-tower Model** [4] A popular model in retrieval task and has been widely introduced to leverage rich content feature.
- **MIND** [6] A recent state-of-the-art model for industrial retrieval task. The number of user interests is set to 4 and 5 for the Meituan dataset and the Amazon dataset, respectively.

We implemented these models by distributed TensorFlow and used Faiss [5] to retrieve the top-N items from the large-scale item pool. The embedding dimension and batch size were fixed to 32 and 256 for all models. All models were trained by the Adam optimizer. To ensure a fair comparison, other hyperparameters of all models were individually tuned to achieve optimal results. For DAT, the number of FC layers in each tower was fixed to 3, with dimensions 256, 128 and 32, respectively. The dimensions of augmented vectors $\mathbf{a}_u$ and $\mathbf{a}_v$ were both set to $d = 32$ while λ_1, λ_2 were set to 0.5 and λ_3 was set 1. To evaluate the offline effectiveness of the various models, we employed HitRate@K and MRR metrics, which were widely used in industrial retrieval. K was set to 50 and 100 as the retrieval module was required to retrieve a relatively large number of candidate items to feed the ranking module. Due to the large scale of testing instances, we employed a scaled version of the MRR by a factor of 10.

[1] https://www.meituan.com/

Table 2: Performance Comparison on two datasets in terms of HitRate and MRR (w/o is short for without).

Models	Meituan			Amazon		
	HR@50	HR@100	MRR	HR@50	HR@100	MRR
WLAS	0.2917	0.4146	0.3375	0.0242	0.0359	0.0351
FM	0.4831	0.6672	0.5012	0.0406	0.0634	0.0589
YouTubeDNN	0.4228	0.5142	0.4512	0.0378	0.0599	0.0524
Two-tower	0.5395	0.7159	0.5472	0.0464	0.0732	0.0625
MIND	0.5507*	0.7327*	0.5843*	0.0490*	0.0784*	0.0673*
DAT	**0.5796**	**0.7682**	**0.6154**	**0.0519**	**0.0836**	**0.0711**
DAT(w/o CAL)	0.5655	0.7512	0.6009	0.0503	0.0816	0.0698

3.3 Offline Results

3.3.1 Model Comparison. The experimental results of DAT and baselines on two datasets are reported in Table 2. The best results are listed in bold, and the best results of baseline are marked with star (*). Clearly, DAT outperformed all baseline models improving HitRate@100 by 4.84% and 6.63% on two datasets over the best baseline. This demonstrates the effectiveness of DAT and further justifies the importance of the Adaptive-Mimic Mechanism (AMM) and the Category Alignment Loss (CAL). WALS, the matrix factorization approach, received poor performance compared with other methods, confirming the effectiveness of deep learning in the retrieval stage of recommender systems. The improvement obtained by FM compared with YouTubeDNN highlights the advantage of feature interaction. Moreover, with the help of deep learning, the two-tower model performed much better than FM and YouTubeDNN, which can be explained by the fact that it can learn query and item representations from more valuable content features. Furthermore, considering that user has multiple interests, MIND performs better than the two-tower model. Finally, the advantage of DAT(w/o CAL) over MIND and the two-tower model and can be attributed to the AMM that can exploit the information interaction between the two towers.

3.3.2 Dimension of Augmented Vectors. The augmented vector in DAT plays a pivotal role in modeling the the information interaction, to analyze the impact of the dimension, we investigated the performance of DAT on the two datasets with respect to the dimensions of augmented vectors. As Figure 2 shows, the performance of DAT on Meituan improved with the increase of dimension while the performance of DAT on Amazon improved in the first place and then downgraded subsequently. This is caused by the difference in the amount of data between the two datasets. In addition, regardless of the dimensionality, it can always achieve better performance, which justify the effectiveness of the augmented vector.

3.4 Online Experiments

In addition to offline studies, we conducted online experiments by deploying DAT to handle the real traffic for one week in a recommender system which serves 60 million users per day. To make a fair comparison, the retrieval stage was followed by the same ranking procedure. CTR and GMV, two widely used industrial metrics, were used to measure the performance of models for serving online traffic. The baseline method for online experiments was the two-tower model, which was the base retrieval algorithm serving the majority of the online traffic. There were one hundred candidate items retrieved by each method and fed to the ranking stage. Figure 3 shows the online results in 7 successive days. Our model outperformed the baseline with a large margin, with an overall average improvements of 4.17% and 3.46% in terms of CTR and GMV, respectively.

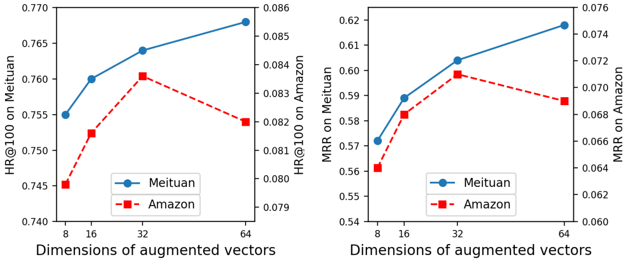

Figure 2: Performance in terms of HR@100 and MRR w.r.t the dimensions of augmented vectors on the two datasets.

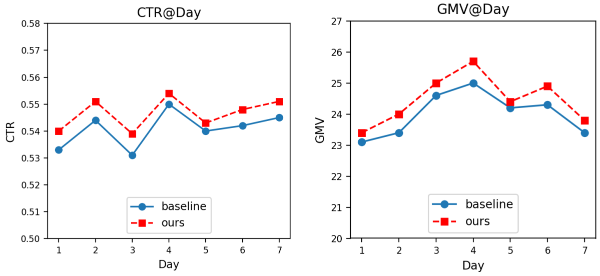

Figure 3: Online performance of DAT and baselines

4 CONCLUSION

In this paper, we present an effective retrieval model named *Dual Augmented Two-tower Model* (DAT) for industrial recommendation systems. It aims to model the information interaction between the two towers and produces better item representations for imbalanced category data. Extensive experiments on offline and online datasets show that the DAT model, with AMM and CAL, can effectively achieve superior performance.

REFERENCES

[1] Christopher R Aberger. 2014. Recommender: An analysis of collaborative filtering techniques. *Personal and Ubiquitous Computing Journal* (2014).

[2] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems.* 191–198.

[3] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web.* 507–517.

[4] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management.* 2333–2338.

[5] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).

[6] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 2615–2623.

[7] Steffen Rendle. 2010. Factorization Machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010.*