Improving Relevance Prediction with Transfer Learning in Large-scale Retrieval Systems

Anonymous Authors[1]

Abstract

Machine learned large-scale retrieval systems require a large amount of training data representing query-item relevance. However, collecting users' explicit feedback is costly. In this paper, we propose to leverage user logs and implicit feedback as auxiliary objectives to improve relevance modeling in retrieval systems. Specifically, we adopt a two-tower neural net architecture to model query-item relevance given both collaborative and content information. By introducing auxiliary tasks trained with much richer implicit user feedback data, we improve the quality and resolution for the learned representations of queries and items. Applying these learned representations to an industrial retrieval system has delivered significant improvements.

1. Introduction

In this paper, we propose a novel transfer learning model architecture for large-scale retrieval systems. The retrieval problem is defined as follows: given a query and a large set of candidate items, retrieve the top-k most relevant candidates. Retrieval systems are useful in many real-world applications such as search (Shen et al., 2014) and recommendation (Covington et al., 2016; Yi et al., 2018; He et al., 2014). The recent efforts on building large-scale retrieval systems mostly focus on the following two aspects:

- *Better representation learning.* Many machine learning models have been developed to learn the mapping of queries and candidate items to a embedding space (Koren et al., 2009; Krichene et al., 2019). These models leverage various features such as collaborative and content information (Wang et al., 2012).
- *Efficient retrieval algorithms.* Given learned representations, efficient algorithms are proposed to retrieve the top-k relevant items given the similarity (distance) metric associated with the embedding space (Broder, 1997; Guo et al., 2016).

However, it is challenging to design and develop real-world large-scale retrieval systems for many reasons:

- *Sparse relevance data.* It is costly to collect users' true opinions regarding item relevance. Often, researchers and engineers design human-eval templates with Likert scale questions for relevance (Chang et al., 2015), and solicit feedback via crowd-sourcing platforms (*e.g.*, Amazon Mechnical Turk).
- *Noisy feedback.* In addition, user feedback is often highly subjective and biased, due to human bias in designing the human-eval templates, as well as the subjectivity in providing feedback.
- *Multi-modality feature space.* We need to learn relevance in a feature space generated from multiple modalities, *e.g.*, query content features, candidate content features, context features, and graph features from connections between query and candidate (Wang et al., 2012; Page et al., 1999; Cui et al., 2010).

In this paper, we propose to learn relevance by leveraging both users' explicit answers on relevance and users' implicit feedback such as clicks and other types of user engagement. Specifically, we develop a transfer-learning framework which first learns the effective query and candidate item representations using a large quantity of users' implicit feedback, and then refines these representations using users' explicit feedback collected from survey responses. The proposed model architecture is depicted in Figure 2.

Our proposed model is based on a two-tower deep neural network (DNN) commonly deployed in large-scale retrieval systems (Krichene et al., 2019). This model architecture, as depicted in Figure 1, is capable of learning effective representations from multiple modalities of features. These representations can be subsequently served using highly efficient nearest neighbor search systems (Guo et al., 2016).

To transfer the knowledge learned from implicit feedback to explicit feedback, we extend the two-tower model by adopting a shared-bottom architecture which has been widely used in the context of multi-task learning (Caruana, 1997).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

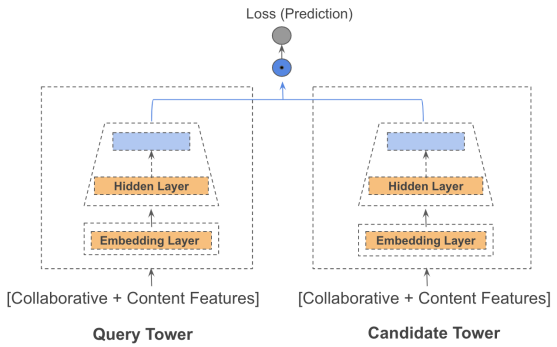

Figure 1. A two-tower DNN model.

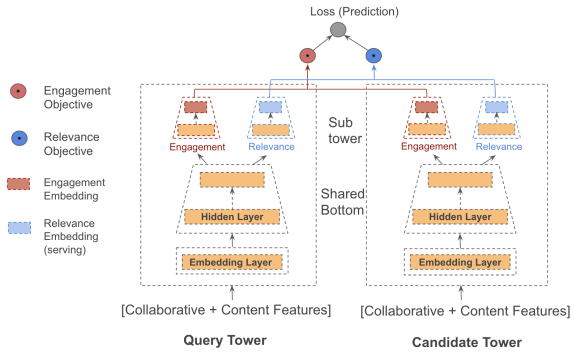

Figure 2. Our proposed two-tower shared-bottom model.

Specifically, the final loss includes training objectives for both the implicit and explicit feedback tasks. These two tasks share some hidden layers, and each task has its own independent sub-tower. At serving time, only the representations learned for explicit feedback are used and evaluated.

Our experiments on an industrial large-scale retrieval system have shown that by transferring knowledge from rich implicit feedback, we can significantly improve the prediction accuracy of sparse relevance feedback.

In summary, our contributions are as follows:

- We propose a transfer learning framework which leverages rich implicit feedback in order to learn better representations for sparse explicit feedback.
- We design a novel model architecture which optimizes two training objectives sequentially.
- We evaluate our model on a real-world large-scale retrieval system and demonstrate significant improvements.

The rest of this paper is organized as follows: Section 2 discusses related work in building large-scale retrieval systems. Section 3 introduces our problem and training objectives. Section 4 describes our proposed approach. Section 5 reports the experimental results on a large-scale retrieval system. Finally, in Section 6, we conclude with our findings.

2. Related Work

In this section, we first introduce some state-of-the-art industrial retrieval systems, and then discuss the application of multi-task learning and transfer learning techniques in retrieval and recommendation tasks.

2.1. Industry-scale Retrieval Systems

Retrieval systems are widely used in large-scale applications such as search (Shen et al., 2014) and recommendation (Covington et al., 2016; Yi et al., 2018; He et al., 2014). In recent years, the industry has moved from reverse index based solutions (Brin & Page, 1998), to machine learned retrieval systems. Collaborative-filtering based systems (Hu et al., 2008; Beutel et al., 2017) have been very popular and successful until very recently, when they were surpassed by various neural network based retrieval models (Liu et al., 2017; Yi et al., 2018; Beutel et al., 2018).

A retrieval system involves two key components: representation learning and efficient indexing algorithms (Manning et al., 2010). Many large-scale industrial retrieval systems have seen success of using two-tower DNN models to learn separate representations for query and candidate items (He et al., 2017; Yang et al., 2018; Krichene et al., 2019).

2.2. Multi-task Learning and Transfer Learning for Retrieval and Recommendation Systems

There has also been work on multi-task retrieval systems for context-aware retrieval applications based on tensor factorization (Zhao et al., 2015). Unfortunatelly, due to limitations on model capacity and serving time constraints, the model cannot be easily adapted to learn complex feature representations from multiple feature sources. Many multi-task DNN based recommendation systems (Covington et al., 2016; Ma et al., 2018) are designed for ranking problems where only a small subset of high quality candidates are scored. These full-blown ranking solutions cannot be easily applied to retrieval problems, where we try to identify thousands of candidates from a large corpus with millions to hundreds of millions of candidate items.

Inspired by these works, we propose a novel framework to combine the benefits of both worlds: (1) the computation efficiency of a two-tower model architecture; and (2) the improved model capability of a multi-task DNN architecture (Caruana, 1997). This enables us to transfer learning from rich implicit feedback to help sparse explicit feedback tasks.

3. Problem Description

In this section, we formalize the retrieval problem, and introduce our training data and training objectives.

3.1. The Retrieval Problem

The retrieval problem is defined as follows. Given a query and a corpus of candidate items, return the top-k relevant items. Let $\{\mathbf{x}_i\}_{i=1}^N \subset \mathcal{X}$ and $\{\mathbf{y}_j\}_{j=1}^M \subset \mathcal{Y}$, respectively, be the feature vectors of queries and candidates in feature space $\mathcal{X}$ and $\mathcal{Y}$, where N and M, respectively, denote the number of queries and candidates. We model the retrieval system as a parameterized scoring function $s(\cdot,\cdot;\boldsymbol{\theta}) : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$, where $\boldsymbol{\theta}$ denotes the model parameters. Items with top-k scores $s(\mathbf{x},\mathbf{y};\boldsymbol{\theta})$ are selected for a given query at inference time. We assume the training data is a set of query and item pairs $\{(\mathbf{x}_t,\mathbf{y}_t)\}_{t=1}^T$, where $\mathbf{y}_t$ is the candidate associated with $\mathbf{x}_t$ which has either explicit or implicit users' feedback, and $T \ll MN$ in practice. Our goal is to fit the scoring function based on these T examples.

3.2. Training with User Feedback

When training a machine learning based retrieval system, the ideal way is to use users' explicit feedback which reflects the relevance of an item to a query. However, asking for users' explicit feedback is costly; hence, many existing systems use implicit feedback from user logs, such as clicks.

In this paper, we study retrieval systems with both explicit and implicit feedback, where implicit feedback is abundant and explicit feedback is relatively sparse.

3.3. Joint Optimization with Auxiliary Objectives

The goal of our retrieval problem is to learn better representations of queries and candidates such that the similarity between a query candidate pair closely approximates relevance. Therefore, our main training objective is to minimize the differences between the predicted relevance and the ground truth.

To facilitate representation learning, we introduce an auxiliary objective which captures user engagement on items, such as clicks of an item, purchase of a product for shopping retrieval, or views of a movie for movie recommendation.

Formally, we aim to jointly learn two objectives $s_{exp}(\cdot,\cdot;\boldsymbol{\theta})$ and $s_{imp}(\cdot,\cdot;\boldsymbol{\theta}')$ while sharing part of the parameters between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. We assume some of the examples $(\mathbf{x}_t,\mathbf{y}_t)$ are in set $\mathcal{E}$ with explicit feedback, and others are in set $\mathcal{I}$ with implicit feedback. In addition, each example $(\mathbf{x}_t,\mathbf{y}_t) \in \mathcal{E}$ is associated with label $l_t \in \mathbb{R}$ representing user' explicit feedback, *e.g.*, response to the relevance survey. Note that $\mathcal{E}$ and $\mathcal{I}$ are not mutually exclusive as some examples can have both implicit and explicit feedback. We use regression loss to fit users' explicit feedback on example set in $\mathcal{E}$. One example loss is the mean squared error (MSE):

$$\mathcal{L}_{exp}(\boldsymbol{\theta};\mathcal{E}) = \frac{1}{|\mathcal{E}|} \sum_{(\mathbf{x}_t,\mathbf{y}_t)\in\mathcal{E}} \left(s_{exp}(\mathbf{x}_t,\mathbf{y}_t;\boldsymbol{\theta}) - l_t\right)^2, \quad (1)$$

where $|\cdot|$ represents the cardinality. On the other hand, we treat the modeling of implicit feedback as a multi-class classification task over the full corpus of items, and use the softmax formulation to model the probability of choosing item $\mathbf{y}$, namely

$$\mathcal{P}(\mathbf{y} \mid \mathbf{x};\boldsymbol{\theta}') = \frac{\exp\left(s_{imp}(\mathbf{x},\mathbf{y};\boldsymbol{\theta}')\right)}{\sum_{j=1}^M \exp\left(s_{imp}(\mathbf{x},\mathbf{y}_j;\boldsymbol{\theta}')\right)}.$$

The maximum likelihood estimation (MLE) can be formulated as

$$\mathcal{L}_{imp}(\boldsymbol{\theta}';\mathcal{I}) = -\frac{1}{|\mathcal{I}|} \sum_{(\mathbf{x}_t,\mathbf{y}_t)\in\mathcal{I}} \log\left(\mathcal{P}(\mathbf{y}_t \mid \mathbf{x}_t;\boldsymbol{\theta}')\right). \quad (2)$$

With loss multipliers w and w', we jointly optimize the losses in (1) and (2) by optimizing

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\theta}') = w \cdot \mathcal{L}_{exp}(\boldsymbol{\theta};\mathcal{E}) + w' \cdot \mathcal{L}_{imp}(\boldsymbol{\theta}';\mathcal{I}).$$

4. Model Architecture

In this section, we describe our proposed framework to learn relevance for large-scale retrieval problems. We extend the two-tower model architecture by introducing a shared-bottom model architecture on both towers.

4.1. Two-tower Model Architecture

Figure 1 provides a high-level illustration of the two-tower DNN model architecture. Given a pair of query and item represented by feature vectors $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}$, respectively, the left and right tower provides two DNN based parameterized embedding functions $u : \mathcal{X} \times \mathbb{R}^d \mapsto \mathbb{R}^k$, $v : \mathcal{Y} \times \mathbb{R}^d \mapsto \mathbb{R}^k$ which encode features of query and item to a k-dimensional embedding space. The scoring function is then computed as the dot product between the query and item embeddings at the top layer, *i.e.*,

$$s(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) = \langle u(\mathbf{x},\boldsymbol{\theta}),\ v(\mathbf{y},\boldsymbol{\theta})\rangle.$$

4.2. Shared-bottom Architecture for Transfer Learning

To enable multi-task learning, we extend the two-tower model by adopting the shared-bottom architecture. Specifically, we introduce two sub-towers on top of the bottom hidden layers, one for the explicit-feedback task and the other for the implicit-feedback task. The outputs of bottom hidden layers are fed in parallel to the two sub-towers. The bottom hidden layers are shared between the two sub-towers (Caruana, 1997), and are referred to as shared-bottom layers. The final model architecture is depicted in Figure 2.

4.3. Training and Serving Schema

During training, we first train the model for the auxiliary user engagement objective, which uses the cross entropy

loss. Having learned the shared representations, we fine-tune the model for the main relevance objective, which uses the squared loss. To prevent potential over-fitting caused by the sparse relevance data, we apply stop gradients for the relevance objective on the shared-bottom layers.

For serving, we only need to store and serve the top layer of the two relevance sub-towers to predict the relevance.

5. Evaluation

In this section, we describe the experiments of our proposed framework on an industrial large-scale retrieval system. We first introduce our experiment setup, and then present our results, followed by discussions.

5.1. Experiment Setup

We conduct our experiments on a large-scale retrieval system used for recommending relevant items. Our system contains several millions of candidate items in the corpus. Our training data contains hundreds of thousands of explicit feedback from relevance survey, and billions of implicit feedback from user logs.

We randomly split the data into 90% for training and 10% for evaluation. Model performance was measured on the eval set by the Root Mean Square Error (RMSE) for relevance prediction. The model was implemented in Tensorflow, of which the output embeddings for queries and candidates were served for retrieval. The hyper-parameters including model size, learning rate, and training steps were carefully tuned for the best model performance.

5.2. Experiment Results

We study the effects of applying transfer learning to relevance prediction. The following experiment results suggest that transfer learning significantly improves prediction quality of sparse relevance task and helps avoid over-fitting.

Table 1 reports relevance RMSE (the lower the better) for different combinations of training objectives and feature types. We see that using implicit feedback leads to a significant improvement as compared to using explicit feedback only. Also, using collaborative information together with content information performs better than the model which uses collaborative information alone.

Table 2 reports relevance RMSE for various model sizes on two sets of training objectives. As a close approximation to the model size, we report the number of multiplications. For models trained with sparse explicit feedback only, increasing the model sizes causes over-fitting and consequently degrades the model performance. In contrast, for models trained with implicit feedback, increasing the model size improves the model performance. This suggests that implicit feedback regularizes the model and prevents over-fitting.

Training Objectives	Feature Used	Relevance RMSE
Explicit Feedback Only	Collaborative Only	0.3583
Explicit Feedback Only	Collaborative + Content	0.3464
Explicit + Implicit Feedback	Collaborative Only	0.2837
Explicit + Implicit Feedback	Collaborative + Content	**0.2673**

Table 1. Eval RMSE on user responses of relevance with different sets of training objectives and feature information.

Training Objectives	Number of Multiplications	Relevance RMSE
Explicit Feedback Only	51K	0.3447
Explicit Feedback Only	68K	0.3464
Explicit + Implicit Feedback	176K	0.2775
Explicit + Implicit Feedback	802K	**0.2673**

Table 2. Eval RMSE on user responses of relevance with varying model sizes.

5.3. Discussions and Future Work

The success of transfer learning hinges on a proper parameterization of both the auxiliary and main tasks. On one hand, we need sufficient capacity to learn a high-quality representation from a large amount of auxiliary data. On the other hand, we want to limit the capacity for the main task to avoid over-fitting to its sparse labels. As a result, our proposed model architecture is slightly different from the traditional pre-trained and fine-tuning model (Hinton & Salakhutdinov, 2006). Besides shared layers, each task has its own hidden layers with different capacities. In addition, we apply a two-stage training with stop gradients to avoid potential issues caused by the extreme data skew between the main task and auxiliary task.

Our experiences have motivated us to continue our work in the following directions:

- We will consider multiple types of user implicit feedback using different multi-task learning frameworks, such as Multi-gate Mixture-of-Expert (Ma et al., 2018) and Sub-Network Routing (Ma et al., 2019). We will continue to explore new model architectures to combine transfer learning with multi-task learning.
- The auxiliary task requires hyper-parameter tuning to learn the optimal representation for the main task. We will explore AutoML (Pham et al., 2018) techniques to automate the learning of proper parameterizations across tasks for both the query and the candidate towers.

6. Conclusion

In this paper, we propose a novel model architecture to learn better query and candidate representations via transfer learning. We extend the two-tower neural network approach to enhance sparse task learning by leveraging auxiliary tasks with rich implicit feedback. By introducing auxiliary objectives and jointly learning this model using implicit feedback, we observe a significant improvement for relevance prediction on a large-scale industrial retrieval system.

References

Beutel, A., Chi, E. H., Cheng, Z., Pham, H., and Anderson, J. Beyond globally optimal: Focused learning for improved recommendations. In *WWW*, 2017.

Beutel, A., Covington, P., Jain, S., Xu, C., Li, J., Gatto, V., and Chi, E. H. Latent cross: Making use of context in recurrent recommender systems. In *WSDM*, 2018.

Brin, S. and Page, L. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

Broder, A. Z. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pp. 21–29. IEEE, 1997.

Caruana, R. Multitask learning. *Machine learning*, 28(1): 41–75, 1997.

Chang, S., Dai, P., Chen, J., and hsin Chi, E. H. Got many labels?: Deriving topic labels from multiple sources for social media posts using crowdsourcing and ensemble learning. In *WWW*, 2015.

Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *RecSys*, 2016.

Cui, B., Tung, A. K., Zhang, C., and Zhao, Z. Multiple feature fusion for social media applications. In *SIGKDD*, 2010.

Guo, R., Kumar, S., Choromanski, K., and Simcha, D. Quantization based fast inner product search. In *Artificial Intelligence and Statistics*, pp. 482–490, 2016.

He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pp. 1–9. ACM, 2014.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *WWW*, 2017.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *science*, 313 (5786):504–507, 2006.

Hu, Y., Koren, Y., and Volinsky, C. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, (8): 30–37, 2009.

Krichene, W., Mayoraz, N., Rendle, S., Zhang, L., Yi, X., Hong, L., Chi, E., and Anderson, J. Efficient training on very large corpora via gramian estimation. In *ICLR*, 2019.

Liu, D. C., Rogers, S., Shiau, R., Kislyuk, D., Ma, K. C., Zhong, Z., Liu, J., and Jing, Y. Related pins at pinterest: The evolution of a real-world recommender system. In *WWW*, 2017.

Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., and Chi, E. H. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *KDD*, 2018.

Ma, J., Zhao, Z., Chen, J., Li, A., Hong, L., and Chi, E. Snr: Sub-network routing for flexible parameter sharing in multi-task learning. *AAAI*, 2019.

Manning, C., Raghavan, P., and Schütze, H. Introduction to information retrieval. *Natural Language Engineering*, 16 (1):100–103, 2010.

Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Pham, H., Guan, M., Zoph, B., Le, Q., and Dean, J. Efficient neural architecture search via parameters sharing. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4095–4104. PMLR, 10–15 Jul 2018.

Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. Learning semantic representations using convolutional neural networks for web search. In *WWW*, 2014.

Wang, J., Zhao, Z., Zhou, J., Wang, H., Cui, B., and Qi, G. Recommending flickr groups with social topic model. *Information retrieval*, 15(3-4):278–295, 2012.

Yang, Y., Yuan, S., Cer, D., Kong, S.-Y., Constant, N., Pilar, P., Ge, H., Sung, Y.-h., Strope, B., and Kurzweil, R. Learning semantic textual similarity from conversations. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pp. 164–174, July 2018.

Yi, X., Chen, Y.-F., Ramesh, S., Rajashekhar, V., Hong, L., Fiedel, N., Seshadri, N., Heldt, L., Wu, X., and Chi, H. Deep retrieval and distributed tensorflow serving. In *SysML Conference*, Feb 2018.

Zhao, Z., Cheng, Z., Hong, L., and Chi, E. H. Improving user topic interest profiles by behavior factorization. In *WWW*, 2015.