Neural Networks 118 (2019) 280–288

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Multi-perspective neural architecture for recommendation system

Han Xiao [a,b,*], Yidong Chen [a], Xiaodong Shi [a], Ge Xu [b]

[a] *Cognitive Science Department, School of Information Science and Engineering, Xiamen University, Xiamen, Fujian 361005, China*
[b] *Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University), China*

ARTICLE INFO

Article history:
Received 9 January 2019
Received in revised form 2 May 2019
Accepted 10 June 2019
Available online 27 June 2019

Keywords:
Recommendation
Neural architecture

ABSTRACT

Currently, there starts a research trend to leverage neural architecture for recommendation systems. Though several deep recommender models are proposed, most methods are too simple to characterize users' complex preference. In this paper, for a fine-grained analysis, users' ratings are explained from multiple perspectives, based on which, we propose our neural architectures. Specifically, our model employs several sequential stages to encode the user and item into hidden representations. In one stage, the user and item are represented from multiple perspectives and in each perspective, the representation of user and that of item put attentions to each other. Last, we metric the output representations from the final stage to approach the users' ratings. Extensive experiments demonstrate that our method achieves substantial improvements against baselines.

1. Introduction

In the era of information explosion, information overload is one of the dilemmas we are confronted with. Recommender systems (RSs) are instrumental to address this problem, because they assist users to identify which information is more preferred (Wei, He, Chen, Zhou, & Tang, 2017; Zhang, He, Liu, Lin, & Stankovic, 2017; Zhang, et al., 2017). Further, to achieve better modeling ability of users' preference, neural architectures that deep learning methods are employed (He, et al., 2017; Xue, Dai, Zhang, Huang, & Chen, 2017). There emerge several latest researches in this trend, such as NeuMF (He, et al., 2017), DMF (Xue et al., 2017) and DeepFM (Guo, Tang, Ye, Li, & He, 2017). Basically, most methods represent user and item in the hidden semantic manner and then metric the hidden representations to predict the ratings by cosine similarity or Multi-layer Perceptron (MLP), Zhang, Yao, and Sun (2017) and Rafeh (2017).

Despite the success of previous methods, they are still too simple to characterize users' complex preference. For the example of movie recommendation, user usually considers the quality of a movie from multiple perspectives, such as acting quality and movie style (Azpiazu, Dragovic, Anuyah, & Pera, 2018). It means that all the perspectives make effects on the preference, which traditional neural methods are difficult to characterize. To tackle this problem, in this paper, we encode user and item into hidden representations from hierarchical multiple perspectives and then metric the hidden representations to predict the preference.

However, there still exist two challenges for the encoding process of multi-perspective modeling: to model hierarchically organized perspectives and to capture the correlation between user and item.

First, the perspectives are hierarchically organized from concrete elements to abstract summarization, shown in Fig. 1. For the example of movie domain (Fig. 1), there are concrete aspects such as actor, director and shooting technique, based on which, abstract aspects such as acting quality and movie style are constructed. In detail, movie style is decided by director and shooting technique, while actor and director mostly determine the acting quality. Regarding the corresponding neural model, the output of each perspective indicates the representation of user/item metric in that perspective. For example, the encoded representation of a user in actor perspective represents the user's preference for actors, while the encoded representation of an item in movie style perspective indicates the style of this movie. Inspired by our analysis, the representation in low-level should support the analysis in high-level, which motivates us to employ a hierarchical neural architecture. Thus, it is reasonable to apply multiple sequential stages and to encode user/item from multiple perspectives in each stage.

Besides, we have studied 24 domains, namely amazon categories,[1] such as CD, pet products, sports, etc. Based on the study, we conclude that all the perspectives from these domains are hierarchically organized, which verifies our motivation.

Second, the correlation between user and item is weak in the encoding process of current models (Xue et al., 2017). However,

* Corresponding author at: Cognitive Science Department, School of Information Science and Engineering, Xiamen University, Xiamen, Fujian 361005, China.

E-mail address: bookman@xmu.edu.cn (H. Xiao).

[1] http://jmcauley.ucsd.edu/data/amazon/.

https://doi.org/10.1016/j.neunet.2019.06.007

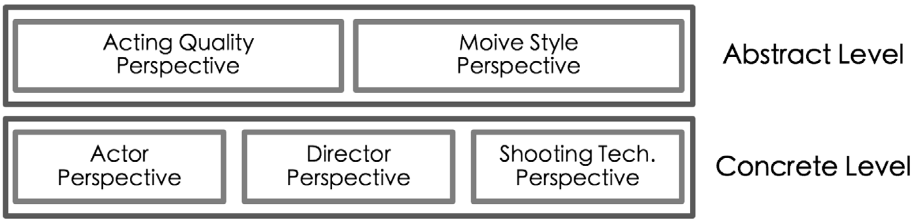


Fig. 1. Hierarchically Organized Perspectives. The representations from concrete level should support the analysis of abstract level.

in fact, from the study of psychology (Bai, 2005; Carlson, Heth, Miller, Donahoe, & Martin, 2009), users' preference is subjective and would be slightly adjusted according to a specific item, while the feature of a specific item could be slightly different from different users' insights. For example, the movie "Titanic" is regarded as romance film for some girls, while it is treated as historical play for some males. Therefore, we employ the attention mechanism (Schmidhuber, 2015) to address the correlated effects between user and item.

In this paper, to model users' complex preference on item, we propose a novel neural architecture for top-N recommendation task. Overall, our model encodes user and item into hidden semantic representations and then metrics the hidden representations into predicted preference degree with cosine similarity. Specifically, regarding the encoding process, our model leverages several sequential stages to model hierarchically organized perspectives. In each stage, there exist several perspectives and in each perspective, the representation for user and that for item would adjust each other by attention mechanism. Besides, we have studied two methods to construct the attention signals, which are listed as "Softmax-ATT" and "Correlated-ATT".

We evaluate the effectiveness of our neural architecture for top-N recommendation task in six datasets from five domains (i.e. movie, book, music, baby product, office product). Experimental results on these datasets demonstrate our model consistently outperforms the other baselines with remarkable improvements and achieves the state-of-the-art performance among deep recommendation models.

In summary, our contributions are outlined as follows:

- We address the importance of hierarchical multi-perspective modeling in recommendation, based on which, we propose novel neural architectures for recommendation systems. Our model focuses on hierarchically organized perspectives and the correlation between user and item. *To our best knowledge, this is the first paper to introduce multi-perspective modeling in neural recommendation system.*
- Experimental results show the effectiveness of our proposed architectures, which outperform other state-of-the-art methods in top-N recommendation task.

The organization of this paper is as follows. First, problem formulation and related work are introduced. Second, our neural architecture is discussed. Third, we conduct the experiments to verify our models. Last, concluding remarks are in the final section.

2. Problem formulation & related work

2.1. Problem formulation

Suppose there are M users $\mathcal{U} = \{u_1, \ldots, u_M\}$ and N items $\mathcal{V} = \{v_1, \ldots, v_N\}$. Let $R \in \mathbb{R}^{M\times N}$ indicate the rating matrix, where its entry R_{ij} is the rating of user i on item j and we denote *unk* if it is unknown. There are two manners to construct the user–item interaction matrix $T \in \mathbb{R}^{M\times N}$, which indicates the user i whether performs operation on the item j as

$$T_{ij} = \begin{cases} 0, & if\ R_{ij}\ is\ unk \\ 1, & otherwise \end{cases} \tag{1}$$

$$T_{ij} = \begin{cases} 0, & if\ R_{ij}\ is\ unk \\ R_{ij}, & otherwise \end{cases} \tag{2}$$

Most traditional models for recommendation system employ Eq. (1) as the input to their models, Wu, Dubois, Zheng, and Ester (2016) and He, et al. (2017), while some latest work takes the known entry that the rating R_{ij} rather than 1 as Eq. (2) shows (Xue et al., 2017). We apply the second setting, because we suppose the explicit ratings in Eq. (2) could reflect the preference level of a user for an item.

The recommendation systems are conventionally formulated as the problem of estimating the rating of each unobserved entry in Y, which is leveraged to rank the items. Model-based approaches that are the mainstream methodology leveraging an underlying model to generate all the ratings:

$$\hat{T}_{ij} = \mathcal{M}(u_i, v_j|\Theta) \tag{3}$$

where $\hat{T}_{ij}$ denotes the predicted score of interaction T_{ij} between user u_i and item v_j, Θ indicates the model parameters and $\mathcal{M}$ denotes the recommendation model that predicts the scores. With the predicted scores by model $\mathcal{M}$, we could rank the items for an individual user to conduct personalized recommendation.

2.2. Neural recommendation systems

First, matrix factorization as semantic latent space methodology is proposed for this task. The classical method of latent factor model (Koren, Bell, & Volinsky, 2009), basically applies the inner product between the hidden representation of user and that of item to predict the entry $\hat{T}_{ij}$ as follows:

$$\hat{T}_{ij} = \mathcal{M}_{LFM}(u_i, v_j|\Theta) = p_i^T q_j \tag{4}$$

where $\hat{T}$ means the predicted score, $\mathcal{M}_{LFM}$ indicates the latent factor model, p_i/q_j, namely the parameter Θ, indicates the hidden representation of user u_i / item v_j, respectively. Also, there are many related researches such as Koren (2008), Mcauley and Leskovec (2013), Bao, Fang, and Zhang (2014).

Then, the extra corpus such as social relationship is incorporated into recommendation for a further improvement, (Ma, Yang, Lyu, & King, 2008). However, because the additional corpus is difficult to obtain and is often full of noise, this methodology is still under limitation.

Last, due to the powerful representation learning ability of neural network, deep learning methods have been successfully applied into this field. Restricted Boltzmann Machines (Salakhutdinov, Mnih, & Hinton, 2007) are the pioneer for this branch.

Meanwhile, autoencoders and denoising autoencoders have also been investigated for this task, (Li, Kawale, & Fu, 2015; Sedhain, Menon, Sanner, & Xie, 2015; Strub & Mary, 2015). The main principle of these methods is to predict users' ratings through learning hidden representations with historical behaviors (i.e. ratings and reviews).

Recently, to learn non-linear interactions, neural collaborative filtering (**NeuCF**) (He, et al., 2017) presents an approach, where users and items are embedded into numerical vectors and then the embeddings are processed by multi-layer perceptron to learn the users' preference. Deep matrix factorization (**DMF**) (Xue et al., 2017) jointly takes the spirit of latent factor model and neural collaborative filtering method. Specifically, DMF independently encodes user and item by multi-layer perceptron (MLP) and then metrics the hidden representation of user and that of item in the manner of Eq. (4) to predict the preference degree. **DeepFM** (Guo et al., 2017) combines the power of factorization machines for recommendation and deep learning for feature learning in a new neural network architecture. *Note, both of DMF and DeepFM achieve the state-of-the-art performance.*

There list the notations used in the following sections. u indicates a user and v indicates an item. i and j are the index for user u and item v, respectively. T denotes the user–item interaction matrix, formulated in Eq. (2), while T^+ denotes the observed interactions, T^- means all the unobservable entries in T and T^-_{sample} denotes the negative instances generated from sampling. Notably, $T^+ \bigcup T^-_{sample}$ means the training and developing dataset, while T^- is the source of testing dataset. Further, we indicate the ith row of matrix T as T_{i*}, jth column as T_{*j} and its (i, j)th entry as T_{ij}.

2.3. Attention mechanism

Attention mechanisms (Vaswani, et al., 2017) in neural networks serve two aspects that orient perception (Kotseruba, Gonzalez, & Tsotsos, 2016) and memory access (Graves, Wayne, & Danihelka, 2014). Latest researches leverage the attention mechanism to filter out the noise and address the task-related features or representations (Yang, He, Gao, Deng, & Smola, 2016). Attention matters because it has been shown to produce the state-of-the-art results in machine translation (Luong, Pham, & Manning, 2015) and other artificial intelligence tasks (Chiu, et al., 2018; Nam, Ha, & Kim, 2017; Zhang, Goodfellow, Metaxas, & Odena, 2018). Besides this technique is one critical component of breakthrough algorithms such as BERT (Devlin, Chang, Lee, & Toutanova, 2018), which has set the new records in accuracy in many tasks. Thus, attention mechanism is part of our best efforts to create better understandings of users' preference in recommendation system (Chen, et al., 2017; Li, et al., 2017; Zhang, Yao, Sun, & Tay, 2019).

2.4. Multi-perspective recommendation system

Multi-perspective technique makes a new branch for recommendation system. Currently, there are few researches regarded with multi-perspective recommendation system. Tavakolifard, et al. (2013) unify temporal, location, and preferential information to provide a more fine-grained recommendation strategy, which leverages explicit features rather than latent semantic analysis. Elkahky, Song, and He (2015) model non-hierarchical several perspectives for recommendation system.

3. Methodology

3.1. Neural architecture

Our neural architecture is demonstrated in Fig. 2. Basically, our model is composed of three components, namely *interaction matrix, sequential stages and cosine similarity.*

Interaction Matrix. *Mentioned in previous section, we form the interaction matrix as Eq. (2), which is the input of our model.* From the interaction matrix T, each user u_i is represented as a high-dimensional vector T_{i*}, which indicates the corresponding user's ratings across all items, while each item v_j is represented as a high-dimensional vector T_{*j}, which means the corresponding item's ratings across all users. Notably, it is a conventional trick to fill the unknown entry as 0, (Xue et al., 2017). To overcome the sparsity of interaction matrix, the input of user and that of item are transformed by linear layer with the activation function ReLU (i.e. $f(x) = max(x, 0)$) as

$$\mathbf{r_u} = \sigma(\mathbf{WT_{i*}} + \mathbf{b_u}) \tag{5}$$

$$\mathbf{r_v} = \sigma(\mathbf{MT}^{\top}_{*\mathbf{j}} + \mathbf{b_v}) \tag{6}$$

where $\mathbf{r_u}/\mathbf{r_v}$ is the output of this layer for user/item, $\mathbf{T}_{i*}/\mathbf{T}_{*j}$ means the input of ith row/jth column-specific interaction matrix for user/item, $\mathbf{W}$, $\mathbf{M}$, $\mathbf{b_u}$, $\mathbf{b_v}$ are the parameters of linear layer and σ is the activation function (i.e. ReLU).

Sequential Stages. *In order to model hierarchically organized perspectives, shown in* Fig. 1, *we leverage multiple sequential stages, shown in* Fig. 2. In each stage, there exist several perspectives to model the representations of user/item from different aspects. In each perspective, the output of last stage is regarded as the input of this perspective, while the outputs of all the perspectives in one stage are concatenated as the output representation of user and item for this stage, respectively, shown in Fig. 2.

Specifically in one perspective, first, *to proceed the information in this perspective,* the inputs of this perspective that the output representations of user and item in the last stage are transformed by linear layer with the activation function *ReLU*.

$$\mathbf{q_{u,s,p}} = \sigma(\mathbf{W_{s,p}}\mathbf{r_{u,s-1}} + \mathbf{b_{u,s,p}}) \tag{7}$$

$$\mathbf{q_{v,s,p}} = \sigma(\mathbf{M_{s,p}}\mathbf{r_{v,s-1}} + \mathbf{b_{v,s,p}}) \tag{8}$$

where σ indicates the ReLU function, $\mathbf{q_{u,s,p}}/\mathbf{q_{v,s,p}}$ is the output for user/item of linear layer of pth perspective in sth stage, $\mathbf{r_{u,s-1}}/\mathbf{r_{v,s-1}}$ is the output for user/item in last stage and $\mathbf{W_{s,p}}$, $\mathbf{M_{s,p}}$, $\mathbf{b_{u,s,p}}$, $\mathbf{b_{v,s,p}}$ are model parameters.

Then, *to consider the correlation between user and item,* attention signal is generated from the output of linear layer by attention mechanism.

$$\mathbf{a_{u,s,p}} = \mathcal{A}_u(\mathbf{q_{u,s,p}}, \mathbf{q_{v,s,p}}) \tag{9}$$

$$\mathbf{a_{v,s,p}} = \mathcal{A}_v(\mathbf{q_{u,s,p}}, \mathbf{q_{v,s,p}}) \tag{10}$$

where $\mathbf{a_{u,s,p}}/\mathbf{a_{v,s,p}}$ is the attention signal for user/item of pth perspective in sth stage and $\mathbf{q_{u,s,p}}/\mathbf{q_{v,s,p}}$ is the output for user/item of linear layer of pth perspective in sth stage. $\mathcal{A}_u/\mathcal{A}_v$ indicates the attention function for user/item, which will be discussed later.

Last, the output of this perspective is generated by weighting the output of linear layer with the attention signal in the manner of element-wise multiplication. Mathematically, we have:

$$\mathbf{r_{u,s,p}} = \mathbf{q_{u,s,p}} \otimes \mathbf{a_{u,s,p}} \tag{11}$$

$$\mathbf{r_{v,s,p}} = \mathbf{q_{v,s,p}} \otimes \mathbf{a_{v,s,p}} \tag{12}$$

where $\mathbf{r_{u,s,p}}/\mathbf{r_{v,s,p}}$ is the output of the pth perspective in sth stage, $\mathbf{a_{u,s,p}}/\mathbf{a_{v,s,p}}$ is the attention signal for user/item of pth perspective

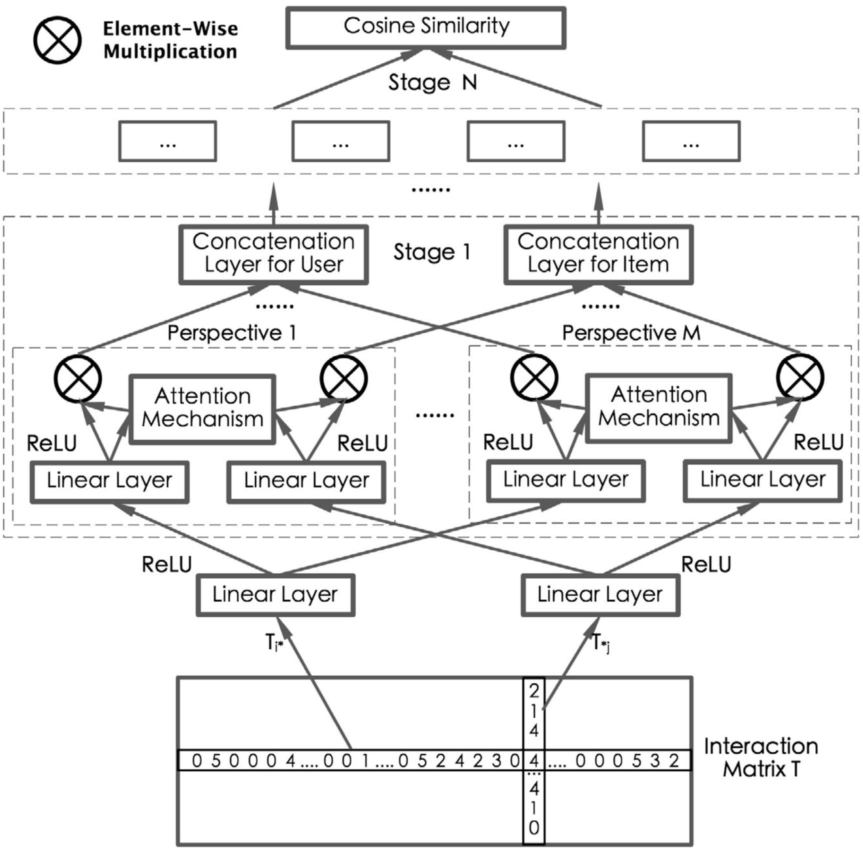


Fig. 2. Proposed Neural Architecture. We leverage the corresponding row/column of interaction matrix as the input of user/item. To characterize hierarchically organized perspectives, we employ several sequential stages to encode the input. In each stage, there exist several perspectives. In each perspective, the input of this perspective, that the output of the last stage, will be encoded into hidden representations by linear transformation with ReLU activation function and then attention mechanism addresses the correlations for the encoded representations of user/item to generate the output of this perspective. Furthermore, the outputs of all the perspectives in one stage are concatenated as the output representation of user and that of item for this stage, respectively. Finally, the representation of user and item would be metric by cosine similarity to predict the user's preference on the item.

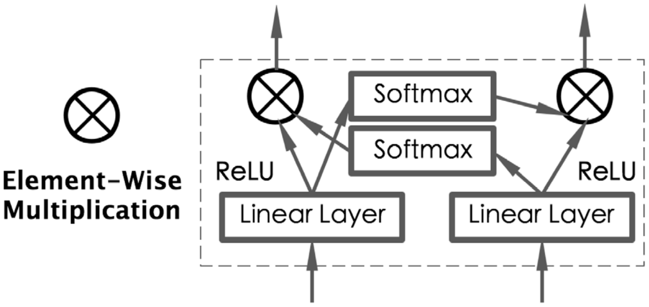


Fig. 3. The first attention mechanism that the "Softmax-ATT", which leverages a simple softmax layer to construct the attention signal.

in sth stage and $\mathbf{q}_{\mathbf{u},\mathbf{s},\mathbf{p}}/\mathbf{q}_{\mathbf{v},\mathbf{s},\mathbf{p}}$ is the output for user/item of linear layer of pth perspective in sth stage. $\otimes$ means the element-wise multiplication.

Cosine Similarity. *To generate the user's u_i preference on the item v_j*, we measure the output representation of user and that of item in the final stage with cosine similarity, which is a conventional operation in neural architecture, (Wang, Mi, & Ittycheriah, 2016), mathematically as

$$\begin{aligned}\hat{T}_{ij} &= cosine(\mathbf{r}_{\mathbf{u},\mathbf{final}}, \mathbf{r}_{\mathbf{v},\mathbf{final}}) \\ &= \frac{{\mathbf{r}_{\mathbf{u},\mathbf{final}}}^{\top}\mathbf{r}_{\mathbf{v},\mathbf{final}}}{\|\mathbf{r}_{\mathbf{u},\mathbf{final}}\| \; \|\mathbf{r}_{\mathbf{v},\mathbf{final}}\|}\end{aligned} \tag{13}$$

where $\hat{T}_{ij}$ is the predicted preference of user u_i on item v_j, $\mathbf{r}_{\mathbf{u},\mathbf{final}}/\mathbf{r}_{\mathbf{v},\mathbf{final}}$ is the output representation of user/item in the final stage, and $\|\cdot\|$ is the length of vector.

3.2. Attention mechanism

Motivated previously, to characterize the correlations between user and item, we leverage attention mechanism to refine the encoded representations of user/item as Eqs. (9) and (10) show. With the attention mechanism, the final representations for user/item are more flexible and more precise to characterize the user's complex preference on the items.

Firstly, shown in Fig. 3, we directly employ a softmax layer to construct the attention signal, which is a conventional and common form for attention-based methods, (Cui, et al., 2016; Yang, Hu, Salakhutdinov, & Cohen, 2017; Yin, Schütze, Xiang, & Zhou, 2015), mathematically as:

$$\mathcal{A}_u(\mathbf{q}_{\mathbf{u},\mathbf{s},\mathbf{p}}, \mathbf{q}_{\mathbf{v},\mathbf{s},\mathbf{p}}) = softmax(\mathbf{A}_{\mathbf{u},\mathbf{s},\mathbf{p}}\mathbf{q}_{\mathbf{v},\mathbf{s},\mathbf{p}}) \tag{14}$$

$$\mathcal{A}_v(\mathbf{q}_{\mathbf{u},\mathbf{s},\mathbf{p}}, \mathbf{q}_{\mathbf{v},\mathbf{s},\mathbf{p}}) = softmax(\mathbf{A}_{\mathbf{v},\mathbf{s},\mathbf{p}}\mathbf{q}_{\mathbf{u},\mathbf{s},\mathbf{p}}) \tag{15}$$

where $\mathbf{A}_{\mathbf{u},\mathbf{s},\mathbf{p}}/\mathbf{A}_{\mathbf{v},\mathbf{s},\mathbf{p}}$ is the attention matrix for user/item of the pth perspective in sth stage, *softmax* is the softmax operation for vector and other symbols are introduced in the last subsection as $\mathcal{A}_u/\mathcal{A}_v$ is the attention function for user/item and $\mathbf{q}_{\mathbf{u},\mathbf{s},\mathbf{p}}/\mathbf{q}_{\mathbf{v},\mathbf{s},\mathbf{p}}$ is the output for user/item of linear layer of pth perspective in sth stage.

Notably, the attention matrices are model parameters to learn. Specifically, the attention signal for user is generated from the representation of item, while the attention signal for item is generated from the representation of user, which accords to our motivation of correlation. We call this attention setting as "Softmax-ATT".

However, *the correlation modeled by simple softmax operation could still be improved.* For more effective correlation modeling, we propose a novel attention structure, shown in Fig. 4. First, we compute the softmax vectors as the first attention method does:

$$\mathbf{a}_{\mathbf{u},\mathbf{s},\mathbf{p}} = softmax(\mathbf{A}_{\mathbf{u},\mathbf{s},\mathbf{p}}\mathbf{q}_{\mathbf{v},\mathbf{s},\mathbf{p}}) \tag{16}$$

$$\mathbf{a}_{\mathbf{v},\mathbf{s},\mathbf{p}} = softmax(\mathbf{A}_{\mathbf{v},\mathbf{s},\mathbf{p}}\mathbf{q}_{\mathbf{u},\mathbf{s},\mathbf{p}}) \tag{17}$$

where $\mathbf{a}_{\mathbf{u},\mathbf{s},\mathbf{p}}/\mathbf{a}_{\mathbf{v},\mathbf{s},\mathbf{p}}$ is the output of softmax layer of pth perspective in sth stage and other symbols are introduced previously. Then, we construct the correlation matrix between the representation of user and that of item, as

$$\mathbf{C}_{\mathbf{s},\mathbf{p}} = \mathbf{a}_{\mathbf{u},\mathbf{s},\mathbf{p}}\mathbf{a}_{\mathbf{v},\mathbf{s},\mathbf{p}}^{\top} \tag{18}$$

where $\mathbf{a}_{\mathbf{u},\mathbf{s},\mathbf{p}}/\mathbf{a}_{\mathbf{v},\mathbf{s},\mathbf{p}}$ is the output of softmax layer, and $\mathbf{C}_{\mathbf{s},\mathbf{p}}$ is the correlation matrix of pth perspective in sth stage, which contains the correlated information of all the dimensions for user/item. Last, we process the correlation matrix with *tanh* activation function and average the row/column as the attention vectors for user/item, as

$$\mathcal{A}_u(\mathbf{q}_{\mathbf{u},\mathbf{s},\mathbf{p}}, \mathbf{q}_{\mathbf{v},\mathbf{s},\mathbf{p}}) = average_{row}(tanh(\mathbf{C}_{\mathbf{s},\mathbf{p}})) \tag{19}$$

$$\mathcal{A}_v(\mathbf{q}_{\mathbf{u},\mathbf{s},\mathbf{p}}, \mathbf{q}_{\mathbf{v},\mathbf{s},\mathbf{p}}) = average_{column}(tanh(\mathbf{C}_{\mathbf{s},\mathbf{p}})) \tag{20}$$

where $average_{row}/average_{column}$ indicates the average operation for row/column and other symbols are introduced previously. *With the explicit computation of correlation matrix, the correlated effects between user and item could be characterized to a better extent (Yin et al., 2015).* We call this attention setting as "Correlated-ATT".

3.3. Explanation & examples

In this subsection, we will discuss the functionalities of each part with the example of movie recommendation. Specifically, we will predict the rating of the user "John" on the movie "Avenger" with the hierarchical perspectives in Fig. 1.

First, for the interaction matrix T that is the input of our system, we take John-specific row that John's ratings on all the movies as the representation of this user, while we take Avenger-specific column that all the users' ratings for Avenger as the representation of this movie.

Second, we proceed the representation of John and Avenger with sequential stages, corresponded in Figs. 1 and 2. There are totally two stages. In the first stage, there are three perspectives namely actor, director and shooting technology. The input of this stage that the initial representations of John and Avenger would be transformed to the new representations in three perspectives respectively with attention mechanism. We concatenate the new representations in three perspectives for John and Avenger as the outputs of this stage. Similarly, in the second stage, there are two perspectives namely acting quality, movie style, which are more abstract. The inputs of second stage are the outputs of first stage that the concatenated representations. In the similar way, the outputs of second stage represent John's preference and abstraction of Avenger.

Last, we leverage the cosine similarity between John's preference and Avenger abstraction to predict John's rating on the movie Avenger.

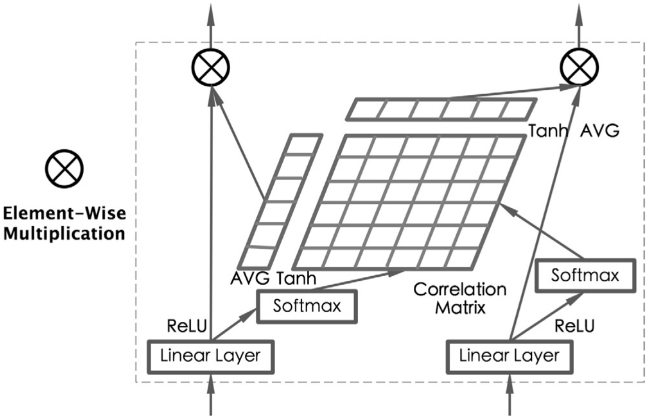


Fig. 4. The second attention mechanism that the "Correlated-ATT", which leverages the correlation matrix to strengthen the correlation characterization.

3.4. Training

The definition of objective function for model optimization is critical for recommendation models. Specifically, regarding our model, we take advantage of point-wise objective function and cross-entropy loss. Actually, though the square loss is largely performed in many existing models, (Hu, Koren, & Volinsky, 2008; Mnih & Salakhutdinov, 2008), neural architectures usually employ cross-entropy loss (He, Liu, Liu, & Zhao, 2017; Wu, Zhang, Yang, Li, & Zhou, 2017). Thus, our objective function $\mathcal{L}$ is as

$$\mathcal{L} = \sum_{(i,j)\in T^{+}\bigcup T^{-}} T_{ij}log\hat{T}_{ij} + (1 - T_{ij})log(1 - \hat{T}_{ij}) \tag{21}$$

where $\mathcal{L}$ is the objective function, T is the golden rating, $\hat{T}$ is the predicted score and other symbols are introduced in Related Work. Specifically as previous literatures (He, Liu, et al., 2017; Xue et al., 2017), the target value T_{ij} is a binarized 1 or 0 for the rating R_{ij}, denoting whether the user u_i has interacted with the item v_j or not. Besides, the model is trained using stochastic gradient descent (SGD) with Adam (Kingma & Ba, 2014), which is an adaptive learning rate algorithm.

The training process needs the negative samples and all the ratings in the training set are the positive ones. Thus, we randomly sample several negative samples that are not in the training/developing/testing dataset for each positive sample. Besides, we apply the concept of negative sample ratio to illustrate how many negative samples would be generated for each positive instance.

4. Experiment

4.1. Experimental setting

Datasets. We evaluate our models on six widely used datasets from five domains in recommender systems: MovieLens 100K (Movie), MovieLens 1M (Movie-1M), Amazon music (Music), Amazon Kindle books (Book), Amazon office product (Office) and Amazon baby product (Baby).[2][3] We process the datasets, according to the previous literatures (He, et al., 2017; Wu et al., 2016; Xue et al., 2017). For the datasets of Movie and Movie-1M, we do not process them, because they are already filtered. Besides, other datasets are filtered to be similar to MovieLens data: only

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

[3] http://jmcauley.ucsd.edu/data/amazon/.

Table 1
NDCG@10 and HR@10 Comparisons of Different Methods. Regarding all the results, we conduct t-test for statistical significance and $p < 0.01$ for all the cases, which means all of the improvements are statistically significant. Soft. represents the Softmax-ATT setting, while Corr. represents the Correlated-ATT setting.

Datasets	Metrics	Baselines			Our Methods	
		NeuMF	DMF	DeepFM	Soft.	Corr.
Movie	NDCG	0.395	0.400	0.400	0.402	**0.410**
	HR	0.670	0.676	0.660	0.686	**0.688**
Movie-1M	NDCG	0.440	0.445	0.421	0.447	**0.452**
	HR	0.722	0.723	0.712	0.732	**0.735**
Book	NDCG	0.477	0.471	0.472	0.483	**0.484**
	HR	0.676	0.667	0.680	0.690	**0.694**
Music	NDCG	0.220	0.230	0.257	0.253	**0.262**
	HR	0.371	0.382	0.391	0.428	**0.445**
Baby	NDCG	0.160	0.162	0.158	0.172	**0.182**
	HR	0.285	0.287	0.281	0.321	**0.366**
Office Product	NDCG	0.233	0.243	0.223	0.261	**0.262**
	HR	0.518	0.520	0.508	0.521	**0.532**

Table 2
Statistics of datasets.

Statistics	#Users	#Items	#Ratings	Density
Movie	994	1.683	100,000	6.294%
Movie-1M	6040	3706	1,000,209	4.468%
Music	1776	12,929	46,087	0.201%
Book	14,803	96,538	627,441	0.004%
Office	941	6679	27,254	4.336%
Baby	1100	8539	30,166	0.321%

those users with at least 20 interactions and those items with at least 5 interactions are retained.[4] We list the statistics of all the six processed datasets in Table 2.

Evaluation. To verify the performance of our model for item recommendation, we adopt the *leave-one-out* evaluation, which has been widely used in the related literatures (He, et al., 2017; Xue et al., 2017). We hold-out the latest interaction as the test item for each user and utilize the remaining dataset for training. Since it is too time-consuming to rank all the items for each user during testing, following He, et al. (2017), Koren et al. (2009) and Xue et al. (2017), we randomly sample 100 items that are not interacted by the corresponding user as the test set for this user. Among the 100 items together with the test item, we get the rank according to the prediction scores. We also use *Hit Ratio (HR)* and *Normalized Discounted Cumulative Grain (NDCG)* to evaluate the ranking performance, (He, Liu, et al., 2017; Xue et al., 2017). As default, in our experiments, we truncate the rank list at 10 for both metrics, where HR/NDCG means HR@10/NDCG@10, as default, as previous literatures (Xue et al., 2017). They are the similar notations for HR@K/NDCG@K. *Note that, to sample 100 items is a conventional setting for this branch of recommendation system and this setting is also taken by NeuMF (He, Liu, et al., 2017) and DMF (Xue et al., 2017).*

Detailed Implementation. We implement our proposed methods based on Tensorflow[5] and the released codes of DMF (Xue et al., 2017). Our codes will be released publicly upon acceptance. To determine the hyper-parameters of our model, we randomly sample one interaction for each user as the developing data and tune hyper-parameters on it. For neural part of our model, we randomly initialize model parameters with Gaussian distribution (with the mean of 0 and the standard deviation of 0.01).

We test the batch size of $[128, 256, 512, 1024]$, the negative instance number per positive instance of $[3, 7, 15]$, the learning rate of $[0.0001, 0.0005, 0.001, 0.005]$, the number of stage $[1, 2, 3]$, the number of perspectives in each stage $[4, 6, 8]$, the dimension of all the linear layers [50, 100, 150], the dimension of the output of non-final stage [50, 100, 150] and the dimension of the output of final stage $[8, 16, 32, 64, 128]$. The optimal settings for our model are listed as: batch size as 256, negative instance number per positive instance as 7, learning rate as 0.0001, number of stage as 3, number of perspectives of each stage as 6, the dimension of all the linear layers as 50, the dimension of the output of non-final stage as 50 and the dimension of the output of final stage as 128.

4.2. Performance verification

Baselines. Since our proposed methods aim to model the relationship between users and items, we follow (He, et al., 2017; Xue et al., 2017) to mainly compare with user–item models. Thus, we leave out the comparison with item–item models, such as CDAE (Wu et al., 2016). Actually, since the neural recommendation methodology just starts to be focused, we list three suitable latest state-of-the-art baseline models, that **NeuMF**, **DMF** and **DeepFM**, which are introduced in Related Work.

Results & Analysis. The comparisons are illustrated in Table 1. Thus, we have concluded as below:

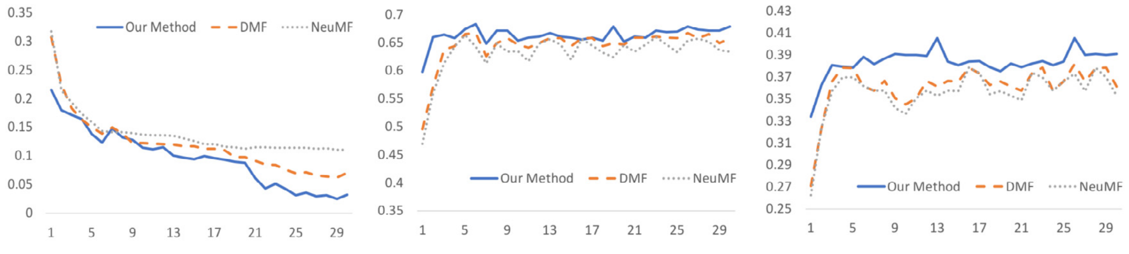
- Our method outperforms the baselines extensively, which justifies the effectiveness of our model.
- "Correlated-ATT" performs better than "Softmax-ATT", which means to characterize the strong correlations between user and item would improve the model performance.
- There exist some domains, where the promotion is obviously larger than the others. We suppose there exist more clear hierarchical perspectives in these domains. For the example of Music domain, there are many low-level aspects such as singer, writer, composer, volume and speed, based on which, high-level aspects such as genre, style, melody are constructed and analyzed.
- We also propose a new setting with "Correlated-ATT", which sets the same parameter number as DMF (Xue et al., 2017). This new setting achieves 0.408 for HR and 0.685 for NDCG on the dataset of Movie. Compared to DMF (Xue et al., 2017), it is concluded that the improvements of our method stem from the model structure.
- Though to sample 100 items as test data (i.e. Conventional Setting) is conventional in this branch of recommendation system (He, et al., 2017; Xue et al., 2017), we still test our method (Correlated-ATT) with all the items as the test data (i.e. Full Setting) for more exact evaluation. In the dataset of Music, conventional setting achieves 0.262 for HR and 0.532 for NDCG, while full setting achieves 0.118 for HR and 0.218 for NDCG. In the dataset of Movie, conventional setting achieves 0.410 for HR and 0.688 for NDCG, while full setting achieves 0.145 for HR and 0.282 for NDCG. In conclusion, the evaluations of conventional setting are in direct proportion to those of full setting, which makes the conventional setting reasonable.

4.3. Sensitive to hyper-parameters

In this subsection, in order to verify the effect of hyper-parameters, we leverage the "Correlated-ATT" setting for attention mechanism over the optimal experimental settings.

[4] We will publish our filtered datasets, once accepted.

[5] https://www.tensorflow.org.



(a) Training Loss (b) HR@10 (c) NDCG@10

Fig. 5. The training loss (averaged over all the training instances), HR@10 and NDCG@10 over iterations on the dataset of Movie.

Table 3
Results for different perspective numbers.

Datasets	Metric	Perspective Number			
		2	4	6	8
Movie	NDCG HR	0.342 0.602	0.361 0.623	0.410 0.688	0.367 0.679
Music	NDCG HR	0.201 0.422	0.259 0.435	0.262 0.445	0.267 0.447
Baby	NDCG HR	0.157 0.310	0.178 0.334	0.182 0.366	0.185 0.369
Office	NDCG HR	0.225 0.496	0.237 0.497	0.262 0.532	0.288 0.544

Table 4
Results for different final latent factor numbers.

Datasets	Metric	Final latent dimension			
		16	32	64	128
Movie	NDCG HR	0.395 0.667	0.400 0.663	0.390 0.687	0.410 0.688
Music	NDCG HR	0.246 0.392	0.248 0.430	0.250 0.433	0.262 0.445
Office	NDCG HR	0.248 0.514	0.273 0.525	0.276 0.523	0.262 0.532
Book	NDCG HR	0.480 0.690	0.480 0.691	0.488 0.692	0.484 0.694

HR@K & NDCG@K. Fig. 6 shows the performance of top-K recommended lists where the ranking position K ranges from 1 to 10. As can be concluded, our method demonstrates consistent improvements over other methods across different K. For the dataset of Movie, our model outperforms DMF by 0.0239 for HR@K and 0.010 for NDCG@K in average, while for the dataset of Music, our method promotes DMF by 0.0360 for HR@K and 0.0261 for NDCG@K in average. This comparison demonstrates the consistent effectiveness of our methods.

Effect of Number of Perspectives. Argued in the previous section, our method takes advantages of multiple perspectives for recommendation. In this experiment, different perspective numbers are tested for the performance variance. From the results in Table 3, we discover that larger perspective numbers could lead to better performance. For the example of Movie, the NDCG@10 improves from 0.342 to 0.367 when the perspective number increases from 2 to 8. In fact, more perspectives could characterize the users' complex preference better, which explains the experimental results.

Effect of Number of Layers/Stages. Note that, we denote one stage as one layer. Since we model hierarchically organized perspectives, the depth or the layer number could be a critical factor in our method. Thus, we conduct experiments to test the effect of depth. Shown in Fig. 7, we could conclude that the 3-layer architectures work best among all the present models. Specifically, on the dataset of Movie, the optimal performance of layer-3 outperforms that of layer-2 by 0.021 for HR@10 and 0.019 for NDCG@10, while on the dataset of Music, the optimal performance of layer-3 improves that of layer-2 by 0.072 for HR@10 and 0.014 for NDCG@10. Thus, we conjecture deeper models could extract more abstract perspectives, which help to boost the performance.

Effect of Final Latent Dimension. Besides the number of perspectives and the number of layers, the final latent dimension is also a sensitive factor, which directly guides the generation of predicted user's preference. We vary the final latent dimension from 16 to 128 for the experiments. Demonstrated in Table 4, we observe that larger final dimension leads to better performance. For the example of Movie dataset, HR@10 increases with latent dimension number. Thus, we suppose larger latent dimension could encode more information into the final results, which could lead to better prediction accuracies.

Training Loss and Performance. Fig. 5 shows the training loss (averaged over all the training instances) and recommendation performance of our method and the state-of-the-art baselines of each iteration on the dataset of Movie. Results on the other datasets show the same trend, thus they are omitted for limited pages. From the results, we could draw two observations. First, we could see that with more iterations, the training loss of our method gradually decreases and the recommendation performance is promoted. The most effective updates are in first 10 iterations and more iterations increase the risk of overfitting, which accords to our common knowledge. Second, our method achieves the lower training loss than DMF, which illustrates that our model could fit the data in a better degree. Thus, a better performance over DMF is expected. Overall, the experiments show the effectiveness of our method.

5. Conclusion

In this paper, we propose a novel neural architecture for recommendation system. Our model encodes user and item from multiple hierarchically organized perspectives with attention mechanism and then metrics the abstract representations to predict user's preference on item. Extensive experiments on several benchmark datasets demonstrate the effectiveness of our proposed methods.

We will publish our poster, slides, datasets and codes at https://www.github.com/....

Regarding the future work, we listed some lines.

1. There exist various attention mechanisms to explore. For example, it is interesting to leverage cosine similarity to construct the attention signals. Besides, the attention signals between perspectives or between stages are valuable to research into. It is a novelty to introduce the cross-layer attention into neural architecture.

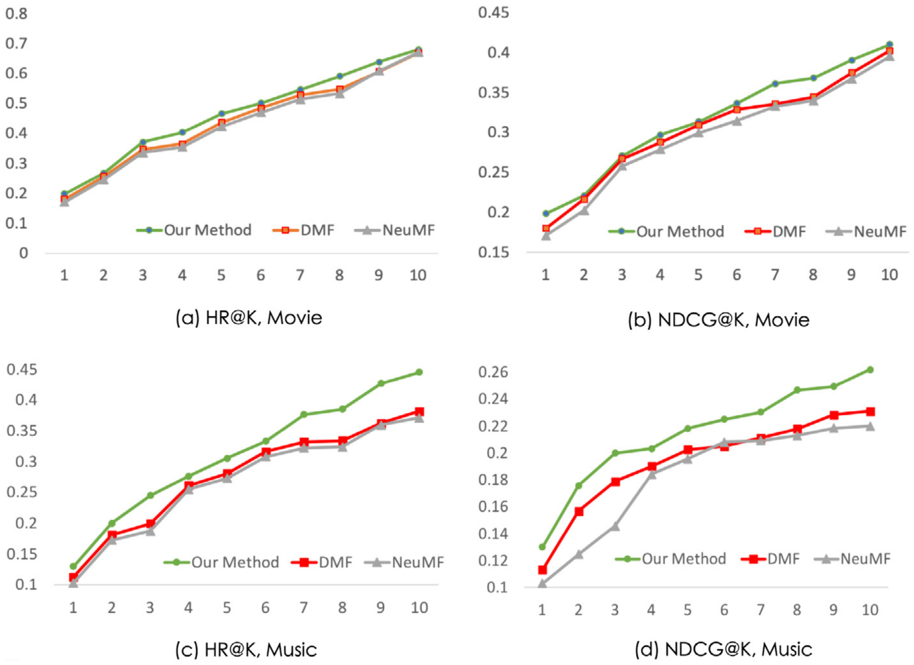


Fig. 6. Evaluation of Top-K item recommendation, where K ranges from 1 to 10 on the datasets of Movie and Music. The y-axis of (a) and (c) is HR@K, while that of (b) and (d) is NDCG@K. The x-axis of all the sub-figures is the K of top-K.

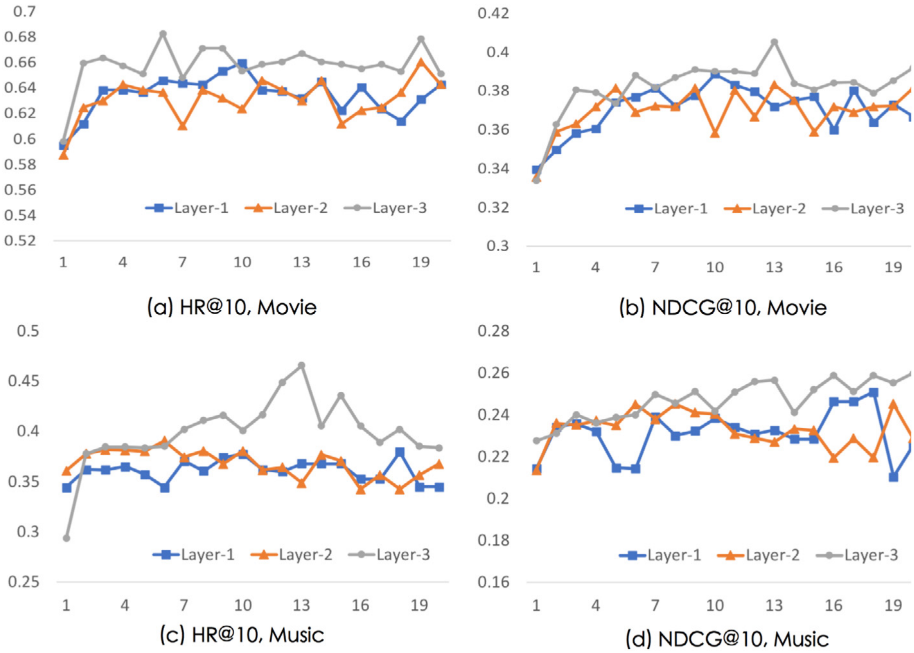


Fig. 7. Results for different layer numbers. The y-axis of (a) and (c) is HR@10, while the y-axis of (b) and (d) is NDCG@10. The x-axis is training epoch.

2. There are abundant review texts, which could be leveraged to enhance the neural recommendation systems. The neural component such as LSTM could be useful for such models. For example, we will encode the review texts into feature vectors with LSTM and then feed the feature vectors into our neural model as the input of neural network.
3. Pairwise objective function is another optional choice for recommendation systems. We will verify our model with pair-wise object functions. Besides, list-wise objective is novel for recommendation, and we could explore the list-wise objective for Top-K recommendation.

Acknowledgment

This work is funded by Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (Grant No. MJUKF-IPIC201804).

References

Azpiazu, Ion Madrazo, Dragovic, Nevena, Anuyah, Oghenemaro, & Pera, Maria Soledad (2018). Looking for the movie seven or sven from the movie frozen? A multi-perspective strategy for recommending queries for children. *Research Gate*, 92–101.

Bai, Rong (2005). The hui minority's sports item choices according to their psychology quality feature. *Journal of Northwest Normal University*.

Bao, Yang, Fang, Hui, & Zhang, Jie (2014). Topicmf: simultaneously exploiting ratings and reviews for recommendation. In *Twenty-eighth AAAI conference on artificial intelligence* (pp. 2–8).

Carlson, Neil R., Heth, Donald, Miller, Harold, Donahoe, John, & Martin, G. Neil (2009). *Psychology: the science of Behavior*. Pearson.

Chen, Jingyuan, Zhang, Hanwang, He, Xiangnan, Nie, Liqiang, Liu, Wei, & Chua, Tat-Seng (2017). Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 335–344). ACM.

Chiu, Chung-Cheng, Sainath, Tara N., Wu, Yonghui, Prabhavalkar, Rohit, Nguyen, Patrick, Chen, Zhifeng, et al. (2018). State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE international conference on acoustics, speech and signal processing* (pp. 4774–4778). IEEE.

Cui, Yiming, Chen, Zhipeng, Wei, Si, Wang, Shijin, Liu, Ting, & Hu, Guoping (2016). Attention-over-attention neural networks for reading comprehension. ArXiv preprint arXiv:1607.04423.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, & Toutanova, Kristina (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv preprint arXiv:1810.04805.

Elkahky, Ali Mamdouh, Song, Yang, & He, Xiaodong (2015). A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th international conference on world wide web* (pp. 278–288). International World Wide Web Conferences Steering Committee.

Graves, Alex, Wayne, Greg, & Danihelka, Ivo (2014). Neural turing machines. ArXiv preprint arXiv:1410.5401.

Guo, Huifeng, Tang, Ruiming, Ye, Yunming, Li, Zhenguo, & He, Xiuqiang (2017). Deepfm: a factorization-machine based neural network for ctr prediction. ArXiv preprint arXiv:1703.04247.

He, Xiangnan, Liao, Lizi, Zhang, Hanwang, Nie, Liqiang, Hu, Xia, & Chua, TatSeng (2017). Neural collaborative filtering. In *25th international world wide web conference* (pp. 173–182).

He, Shizhu, Liu, Cao, Liu, Kang, & Zhao, Jun (2017). Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers), (Vol. 1)* (pp. 199–208).

Hu, Yifan, Koren, Yehuda, & Volinsky, Chris (2008). Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE international conference on* (pp. 263–272). Ieee.

Kingma, Diederik P., & Ba, Jimmy (2014). Adam: A method for stochastic optimization. ArXiv preprint arXiv:1412.6980.

Koren, Yehuda (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 426–434).

Koren, Yehuda, Bell, Robert, & Volinsky, Chris (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Kotseruba, Iuliia, Gonzalez, Oscar J. Avella, & Tsotsos, John K. (2016). A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications. (pp. 1–74). ArXiv preprint arXiv:1610.08602.

Li, Sheng, Kawale, Jaya, & Fu, Yun (2015). Deep Collaborative Filtering via Marginalized Denoising Auto-encoder. In *ACM international on conference on information and knowledge management* (pp. 811–820).

Li, Jing, Ren, Pengjie, Chen, Zhumin, Ren, Zhaochun, Lian, Tao, & Ma, Jun (2017). Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on information and knowledge management* (pp. 1419–1428). ACM.

Luong, Minh-Thang, Pham, Hieu, & Manning, Christopher D. (2015). Effective approaches to attention-based neural machine translation. ArXiv preprint arXiv:1508.04025.

Ma, Hao, Yang, Haixuan, Lyu, Michael R., & King, Irwin (2008). Sorec:social recommendation using probabilistic matrix factorization. In *Acm Conference on Information and Knowledge Management* (pp. 931–940).

Mcauley, Julian, & Leskovec, Jure (2013). Hidden factors and hidden topics:understanding rating dimensions with review text. In *ACM conference on recommender systems* (pp. 165–172).

Mnih, Andriy, & Salakhutdinov, Ruslan R. (2008). Probabilistic matrix factorization. In *Advances in neural information processing systems* (pp. 1257–1264).

Nam, Hyeonseob, Ha, Jung-Woo, & Kim, Jeonghee (2017). Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 299–307).

Rafeh, Reza (2017). Recommender systems in ecommerce. ArXiv.

Salakhutdinov, Ruslan, Mnih, Andriy, & Hinton, Geoffrey (2007). Restricted Boltzmann machines for collaborative filtering. In *International conference on machine learning* (pp. 791–798).

Schmidhuber, J. (2015). *Deep learning in neural networks*. Elsevier Science Ltd.

Sedhain, Suvash, Menon, Aditya Krishna, Sanner, Scott, & Xie, Lexing (2015). AutoRec: Autoencoders meet collaborative filtering. In *International conference on world wide web* (pp. 111–112).

Strub, Florian, & Mary, Jérémie (2015). Collaborative filtering with stacked denoising autoencoders and sparse inputs. ArXiv.

Tavakolifard, Mozhgan, Gulla, Jon Atle, Almeroth, Kevin C., Ingvaldesn, Jon Espen, Nygreen, Gaute, & Berg, Erik (2013). Tailored news in the palm of your hand: a multi-perspective transparent approach to news recommendation. In *Proceedings of the 22nd international conference on world wide web* (pp. 305–308). ACM.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Wang, Zhiguo, Mi, Haitao, & Ittycheriah, Abraham (2016). Semi-supervised clustering for short text via deep representation learning. In *The 20th SIGNLL conference on computational natural language learning*.

Wei, Jian, He, Jianhua, Chen, Kai, Zhou, Yi, & Tang, Zuoyin (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, *69*, 29–39.

Wu, Yao, Dubois, Christopher, Zheng, Alice X., & Ester, Martin (2016). Collaborative denoising auto-encoders for top-n recommender systems. In *ACM international conference on web search and data mining* (pp. 153–162).

Wu, Shuangzhi, Zhang, Dongdong, Yang, Nan, Li, Mu, & Zhou, Ming (2017). Sequence-to-dependency neural machine translation. In *Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers), (Vol. 1)* (pp. 698–707).

Xue, Hong Jian, Dai, Xin Yu, Zhang, Jianbing, Huang, Shujian, & Chen, Jiajun (2017). Deep matrix factorization models for recommender systems. In *International joint conference on artificial intelligence*, (pp. 3203–3209).

Yang, Zichao, He, Xiaodong, Gao, Jianfeng, Deng, Li, & Smola, Alex (2016). Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 21–29).

Yang, Zhilin, Hu, Junjie, Salakhutdinov, Ruslan, & Cohen, William W. (2017). Semi-supervised QA with generative domain-adaptive nets. ArXiv preprint arXiv:1702.02206.

Yin, Wenpeng, Schütze, Hinrich, Xiang, Bing, & Zhou, Bowen (2015). Abcnn: Attention-based convolutional neural network for modeling sentence pairs. ArXiv preprint arXiv:1512.05193.

Zhang, Han, Goodfellow, Ian, Metaxas, Dimitris, & Odena, Augustus (2018). Self-attention generative adversarial networks. ArXiv preprint arXiv:1805.08318.

Zhang, Desheng, He, Tian, Liu, Yunhuai, Lin, Shan, & Stankovic, John A. (2017). A Carpooling recommendation system for taxicab services. *IEEE Transactions on Emerging Topics in Computing*, *2*(3), 254–266.

Zhang, Daqiang, Hsu, Ching Hsien, Chen, Min, Chen, Quan, Xiong, Naixue, & Lloret, Jaime (2017). Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems. *IEEE Transactions on Emerging Topics in Computing*, *2*(2), 239–250.

Zhang, Shuai, Yao, Lina, & Sun, Aixin (2017). Deep learning based recommender system: A survey and new perspectives. *Arxiv*.

Zhang, Shuai, Yao, Lina, Sun, Aixin, & Tay, Yi (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, *52*(1), 5.