

Research Article

An End-to-End Review-Based Aspect-Level Neural Model for Sequential Recommendation

Yupeng Liu ¹, Yanan Zhang,¹ and Xiaochen Zhang ²

¹Department of Software, Harbin University of Science and Technology, Xuefu Street 52, Harbin 150001, China

²School of Materials Chemical and Engineering, Heilongjiang Institute of Technology, Harbin, China

Correspondence should be addressed to Xiaochen Zhang; zxc161616@126.com

Received 28 November 2020; Revised 3 February 2021; Accepted 9 February 2021; Published 20 February 2021

Academic Editor: Stefania Tomasiello

Copyright © 2021 Yupeng Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Users' reviews of items contain a lot of semantic information about their preferences for items. This paper models users' long-term and short-term preferences through aspect-level reviews using a sequential neural recommendation model. Specifically, the model is devised to encode users and items with the aspect-aware representations extracted globally and locally from the user-related and item-related reviews. Given a sequence of neighbor users of a user, we design a hierarchical attention model to capture union-level preferences on sequential patterns, a pointer model to capture individual-level preferences, and a traditional attention model to balance the effects of both union-level and individual-level preferences. Finally, the long-term and short-term preferences are combined into a representation of the user and item profiles. Extensive experiments demonstrate that the model substantially outperforms many other state-of-the-art baselines substantially.

1. Introduction

In the era of information explosion, information overload is an important problem faced by users. Recommendation system came into being to solve this problem. They have evolved into a fundamental tool to help users in selecting items. Recommendation models include review-based model [1], attention-based model [2–4], aspect-based model [5], etc [6–8].

In the academic research area, a lot of work is focused on modelling long-term preferences. However, there is no appropriate model for short-term preferences from near-neighbor users. Near-neighbor users often influence the item purchase decisions made by a user. For example, a user would be highly interested in buying Apple mobile phones if his/her near-neighbor users buy them. Recently, sequential recommendations have drawn an increasing attention from both academic and industrial circles. The task is to identify an item purchased by a user by considering his or her temporal preference as a sequence.

Meanwhile, many techniques have been developed to tackle the data sparsity problem by utilizing the semantic

signals provided by reviews. To model the rich semantic information in textual reviews, it is necessary to move beyond the surface-level feature representations of words in reviews. Consider the following two short sentences which contain the word “short”: (1) “a short battery life,” and (2) “a short loading time.” Here, “short” has different sentimental information in the two sentences. Every word has different meanings in different contexts. Word embedding only considers words without considering the different meanings of each word in different contexts.

To this end, we propose an end-to-end review-based aspect-level neural model (RANM) for a sequential recommendation. Compared with the pipeline model, our end-to-end model overcomes the inherent problems of the pipeline model: time delay, parameter redundancy, and error propagation. The incorporation of reviews provides more abundant semantic information and enhances the model's expressive ability. Specifically, we embed user-related and item-related reviews into a continuous low-dimensional dense vector space by utilizing aspect-aware convolution and self-attention. Aspect-aware convolution can obtain semantic-oriented sentimental information in

reviews locally, while self-attention can obtain word association information in reviews globally. The two sides are just complementary. Then, the hierarchical attention layer is used to obtain the sequence pattern of neighbor users at the union level, and pointer mechanism is used at the individual level. Attention mechanism is used to balance the effects of the two sides. The hierarchical attention layer can obtain a multigranularity union-level neighbor sequence pattern, whereas the point mechanism can obtain an individual-level neighbor sequence pattern. The two sides are just complementary. Finally, we combine the long-term and short-term preferences to obtain a hybrid representation method for more accurate item recommendations. Overall, the key contributions of this paper towards sequential recommendation are summarized as follows:

- (i) In this paper, recommendation reviews are used to precisely represent users and items. Our model can alleviate the data sparsity problem and provide a good interpretability for recommendation tasks. To the best of our knowledge, this is the first paper to harness the rich semantic information from neighbor users' reviews in the neural sequential recommendation system.
- (ii) More fine-grained user and item profiles are developed. Furthermore, we use an aspect-aware convolution and a self-attention layer to identify users, their neighbors, and item representation. Meanwhile, we introduce a hierarchical attention mechanism and a review pointer to model multigranularity union-level and individual-level user preferences.
- (iii) We conduct extensive experiments on a large number of datasets (Amazon and Yelp) to verify the effectiveness of the model. The importance of different components of the model and the sensitivity of the parameters are investigated. It is compared with many classic baseline systems and achieves good results on sparse datasets.

The remainder of this paper is organized as follows. In Section 2, we discuss the existing work related to our method. In Sections 3 and 4, we describe the details of RANM, model architecture, and training method. Section 5 describes our experiment setup and compares our proposed model with many state-of-the-art baselines. Finally, we conclude this paper in Section 6.

2. Related Work

2.1. Review-Based Recommendation. In order to improve the accuracy and interpretability of recommendation, there is a lot of research on utilizing reviews. Both convolution neural network (CNN) [9] and recurrent neural network (RNN) [10] have been widely adopted to extract semantic representation from reviews for rating prediction [1–4, 11–14]. DeepCoNN [13] is the first attempt to jointly model both the user and the item from reviews using neural networks. From the perspective of multimodel combination, TransNets [1]

extend the DeepCoNN [13] by introducing an additional layer to represent the user-item pair. The target network assists the source network in scoring predictions. Wu et al. [11] proposed a model to exploit user-item interaction features from auxiliary users' reviews. Wu et al. [12] modelled a joint representation for a given user and item pair, which includes review-based and interaction-based feature learning. From the discovery of multiple preferences, Tay et al. [4] proposed a multipointer model that can combine multiple views of user-item interactions through review-level and word-level co-attention. Li et al. [14] considered a viewpoint of a user on an item as a semantic representation unit, which is organized into multiple logic interest units.

These methods can be classified as collaborative filtering based on interaction. From single-model view, these all are late interaction models. Our method includes not only the early interaction between neighbor users and items but also the late interaction between users and items. Our model includes a combination of the neighbor model and the user model and has a variety of preferences, such as long-term preferences and union-level and individual-level short-term preferences.

2.2. Topic-Based Recommendation. Many research studies try to extract semantic features from texts through topic model [15, 16]. CTR [17] assumes that the latent factors of items depend on the latent topic distributions of their text. TopicMF [18] links the latent topics and latent factors by using a defined transform function. McAuley et al. [19] integrated the topic method into matrix factorization through corpus likelihood regularization. Xu et al. [20] presented a model combining rating prediction and topic selection, which incorporates reviews and co-clusters of hidden user communities and item groups.

There are not only topic information but also sentimental information in reviews. Some solutions model latent factors into the framework of topic graphical model. Wang et al. [21] presented a latent aspect rating analysis model to determine the relative importance of a topical aspect. Diao et al. [22] jointly modelled ratings and review generation through aspects and their sentiments. Other solutions incorporate topical factors learned from reviews into a latent factor learning framework. Cheng et al. [23] jointly modelled the aspect-level importance and rating of reviews and item images, which were put into matrix decomposition framework for rating prediction. Shao et al. [24] matched users' and items' sentiment-aware multimodal topic models. Compared with the first kind of method, this model can be optimized both jointly and independently.

2.3. Aspect-Based Recommendation. Aspect-based recommendation systems can be divided into two main categories. The first category uses external sentiment analysis tools to extract the aspects. Zhang et al. [25] developed a multimatrix factorization model using a user-item rating matrix, a user-feature attention matrix, and an item-feature quality matrix. Chen et al. [3] learned to rank user preferences based on a phrase-level sentiment analysis across multiple categories

and further integrated this framework with matrix factorization at both the item and category levels. SLUM [26] predicts the sentiment information and then identifies the most valuable aspects of the user on that item. These works rely on the performance of the external sentiment analysis toolkit.

The second category automatically obtains the aspects of reviews by an embedded model component. Chin et al. [5] modelled the multifaceted process behind how users rate items by estimating the aspect-level user and item importance based on the neural co-attention mechanism. A³NCF [27] defines aspects in reviews as a combination representation of topic information and embedding information. Li et al. [28] used a special CNN network to capture aspect-aware representations in reviews. Compared with the first kind of method, this explicit aspect extraction method causes error accumulation in the downstream recommendation task.

Topic-based and aspect-based recommendations are shallow latent semantic methods, but our model is a deep latent semantic method through complex hierarchical structure. In addition, our approach has a more unified model representation for joint training and inference, and the representation of latent semantics is also more diverse.

2.4. Sequential Recommendation. Sequential recommendation is an important means to use implicit feedback. Sequential pattern learning is widely verified as a critical issue in sequential recommendation [29–32]. Rendle et al. [29] presented a factorized personalized Markov chain model that subsumes both a common Markov chain and a matrix factorization model. Wang et al. [30] implemented a hierarchical representation model that can well capture both users' general tastes and sequential behaviors by involving transaction and user representations in a rating prediction. He McAuley [31] integrated similarity-based methods with Markov chains. These methods use a Markov chain to model the sequence patterns. Chen et al. [32] proposed a more flexible model to integrate collaborative filtering with a memory network.

Session/cookie-based recommendation, which does not contain user identification information, is very similar to the sequential recommendation. Hidasi et al. [33] proposed a GRU model for session recommendations. This is generally believed to be the first session recommendation based on a deep neural network (DNN). Tan et al. [34] proposed two practical techniques, namely, data augmentation and distribution shifts, to improve model performance. Jannach and Ludewig [35] proposed a heuristic near-neighbor framework for sessions that are complementary with GRU4REC [33]. Ludewig and Jannach [36] proposed an effective session-based matrix factorization, which presents the results of an in-depth performance comparison of a number of such algorithms, using a variety of datasets and evaluation measures. Our model uses an attention-based mechanism to obtain a more expressive relationship in and out of the sequence.

3. Proposed Model

In this section, we propose an end-to-end review-driven aspect-level neural recommendation model. Figure 1 illustrates the overall network architecture of the user representation network.

3.1. Model Architecture. In Table 1, let \mathcal{U} and \mathcal{I} represent the user and item sets, respectively. Each item $i \in \mathcal{I}$ is associated with a sequence of neighbor users arranged chronologically as $N_u = (N_u^1, \dots, N_u^M)$, where N_u^j is a neighbor user with which the item i has interacted. We merge the user u 's and the item i 's review sets to form user document $D_u = (w_1, \dots, w_{|D_u|})$ and item document $D_i = (w_1, \dots, w_{|D_i|})$, where $|D_u|$ and $|D_i|$ are the user u 's and the item i 's document lengths in number of words, respectively.

Firstly, the user review D_u is transformed into an embedding matrix $\mathbf{M}_u \in R^{|D_u| \times d}$ via a neural embedding layer, where d is the dimension size of the representation vector of each word in the review. Specifically, the neural embedding layer performs a look-up operation through a shared word-embedding matrix. The word-embedding matrix can be initialized using a pretrained word vector [37, 38].

The same color is used for the same type of review and review-based vector representation in Figure 1, such as for users' reviews and review-based vectors. The same color is used for the same type of subnetwork, such as the self-attention network, CNN, and attention network.

3.2. Finding Neighbor Users. Finding neighbor users can help in determining the neighbor users in a social network. We use neighbor users' reviews to improve the user representation model, since they can provide more information for the current user. Here a neighbor user's preference is similar to that of the current user. The key here is to determine the current user's neighbor users. This paper adopts a matrix factorization method [39], a standard method to detect neighbor users. The basic idea is to decompose the rating matrix to obtain the representation of hidden variables, so as to obtain the neighbor users through a similarity function.

Specifically, the user's ratings on the item are expressed by a matrix $\mathbf{M} \in R^{|\mathcal{U}| \times |\mathcal{I}|}$, where the number of elements in the user set \mathcal{U} and item set \mathcal{I} is denoted by $|\mathcal{U}|$ and $|\mathcal{I}|$. Three matrices are used to approximate the rating matrix. In the learning algorithm, we use an iteration optimization method for SVD matrix factorization.

$$\min_{\mathbf{F} \geq 0, \mathbf{G} \geq 0, \mathbf{S} \geq 0} \|\mathbf{M} - \mathbf{F}\mathbf{S}\mathbf{G}^T\|, \text{ s.t. } \mathbf{F}^T\mathbf{F} = \mathbf{I}, \mathbf{G}^T\mathbf{G} = \mathbf{I}, \quad (1)$$

where $\mathbf{F} \in R^{|\mathcal{U}| \times K}$ represents the posterior probability of K topic clusters for each item, $\mathbf{S} \in R^{K \times K}$ represents the distribution of each topic k , and $\mathbf{G} \in R^{K \times |\mathcal{I}|}$ represents the posterior probability of k topic for each user.

We use matrix factorization to obtain the posterior probability. To infer \mathbf{F} , \mathbf{S} , and \mathbf{G} , the solution to the problem is obtained by the following updating rule:

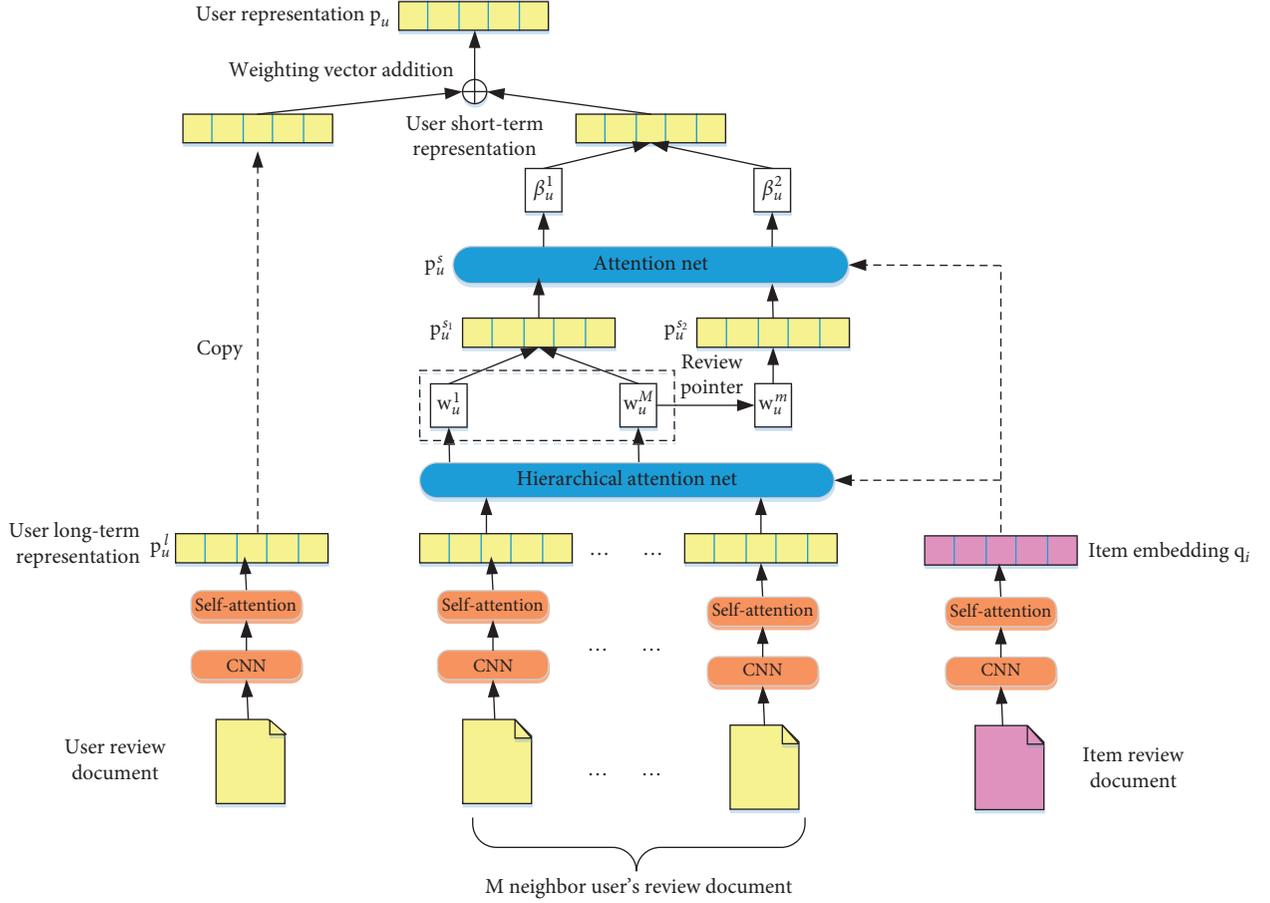


FIGURE 1: Network architecture of RANM.

TABLE 1: Notations and their definitions.

Notation	Definition
\mathcal{U}	User set
\mathcal{F}	Item set
$u \in \mathcal{U}$	User
$i \in \mathcal{F}$	Item
D_u	User u 's document
D_i	Item i 's document
\mathbf{q}_i	Item i 's representation
\mathbf{p}_u^l	User u 's long-term preference representation
\mathbf{p}_u^{s1}	User u 's union-level short-term preference representation
\mathbf{p}_u^{s2}	User u 's individual-level short-term preference representation
\mathbf{p}_u^s	User u 's short-term preference representation
\mathbf{q}_u^j	User u 's j -th neighbor user representation
\mathbf{p}_u	User u 's overall preference representation
Θ_{ac}	Parameter set in CNN net
Θ_{sa}	Parameter set in self-attention net
Θ_{ha}	Parameter set in hierarchical attention net
Θ_{an}	Parameter set in attention net
r_{ui}	Rating from user u for item i

$$\mathbf{F}_{ik} \leftarrow \mathbf{F}_{ik} \frac{(\mathbf{MGS}^T)_{ik}}{(\mathbf{FF}^T \mathbf{MGS}^T)_{ik}}, \quad (2)$$

$$\mathbf{S}_{kk'} \leftarrow \mathbf{S}_{kk'} \frac{(\mathbf{F}^T \mathbf{MG})_{kk'}}{(\mathbf{F}^T \mathbf{FSG}^T \mathbf{G})_{kk'}}, \quad (3)$$

$$\mathbf{G}_{k'j} \leftarrow \mathbf{G}_{k'j} \frac{(\mathbf{M}^T \mathbf{FS})_{k'j}}{(\mathbf{GG}^T \mathbf{M}^T \mathbf{FS})_{k'j}}. \quad (4)$$

The similarity of users u and v , given the rating matrix \mathbf{F} , is computed using the Pearson correlation coefficient as follows:

$$\text{sim}(u, v) = \frac{\sum_{k=1}^K (\mathbf{F}_{uk} - \bar{\mathbf{F}}_u)(\mathbf{F}_{vk} - \bar{\mathbf{F}}_v)}{\sqrt{\sum_{k=1}^K (\mathbf{F}_{uk} - \bar{\mathbf{F}}_u)^2} \sqrt{\sum_{k=1}^K (\mathbf{F}_{vk} - \bar{\mathbf{F}}_v)^2}}, \quad (5)$$

where $\bar{\mathbf{F}}_u$ and $\bar{\mathbf{F}}_v$ denote the average weighting ratings of users u and v , respectively. The process for neighboring items is the same as for the neighbor users. Similar to equation (5), we can compute the similarity between two item vectors using a topic-item matrix \mathbf{G} .

3.3. Aspect-Level CNN. In this section, we describe the aspect-level convolution process of user reviews for extracting aspect-level information from reviews. Traditionally, we use the same word-embedding representation for the same word. However, the semantics or sentiment polarity of the same word can be different for different aspects. In this paper, an aspect-based transformation matrix $\mathbf{T}_a \in R^{d \times d}$ is used to transform a word-embedding matrix \mathbf{M}_u into an aspect-based word-embedding matrix $\mathbf{M}_u^a = \mathbf{M}_u \mathbf{T}_a$. The word embedding for K aspects can be expressed as a tensor $\mathbf{M}_u \in R^{|D_u| \times d \times K}$. Different from the traditional word-embedding representation, this paper presents the different aspects of each word.

Then, motivated by the work of Li et al. [28], we use a CNN to encode the current word's context. The tensor is similar to the multichannel feature representation of an image. In this paper, the classical image feature processing method is used to perform the convolution operation on an aspect-based feature embedding tensor. Specifically, there are n filters/kernels $\mathbf{F}^k \in R^{h \times d \times K}$, $1 \leq k \leq n$, where h is the height of the filter. For instance, if $n = 10$, $h = \{1, 3, 5, 7, 9\}$, then there will be two filters for each size. To obtain the nonlinear feature transformation, we use ReLU as the activation function after the convolution operation using n filters.

$$\mathbf{c}_i^k = \text{ReLU}\left(\mathbf{M}_u \left[i - \frac{h}{2} : i + \frac{h}{2} \right] \circ \mathbf{F}^k + b^k\right), \quad (6)$$

where $\mathbf{M}_u [i - h/2 : i + h/2]$ is the dimension h extracted from the first order of the third-order tensor, \circ denotes the elementwise product of the tensor and the k -th convolution kernel \mathbf{F}^k , and $1 \leq i \leq l$ is the starting position of the sliding window. Specifically, the context uses a window spanning

$h/2$ words on both sides of the current word. The result is $[\mathbf{c}_1, \dots, \mathbf{c}_l]$ ($\mathbf{c}_i \in R^n$ is the context feature for the word, and n is the number of filters). Because different convolution kernels have different convolution ranges, the dimensions of the generated vectors are different. For processing convenience, different filling sizes are used for convolution kernels of different sizes. The parameter set Θ_{ac} of the aspect-level CNN layer contains \mathbf{M}_u , \mathbf{T}_a , \mathbf{F}^k , and b^k .

3.4. Self-Attention. In general, a pooling layer is used to refine features after the convolution layer. However, not all words are equally crucial for each user/item. Convolution considers local features, while self-attention takes global features into account. The integration of the two feature extractions can be better considered from global and local views [40]. For better generalization ability, we use the general self-attention [41].

$$\mathbf{c}_i = \sum_j \mathbf{c}_j \delta\left(\frac{\mathbf{c}_i \mathbf{W}_a \mathbf{c}_j^T}{\sqrt{n}}\right), \quad (7)$$

where \mathbf{c}_i and \mathbf{c}_j are the representation vectors of the current word, \mathbf{W}_a are learnable parameters, and δ is a softmax function. Intuitively, the self-attention layer computes the average of all value vectors \mathbf{c}_j , where the weight is a matrix-weighted dot product between the query vector \mathbf{c}_i and the key vector \mathbf{c}_j . A scaling factor n is used to avoid a large weight because of its high dimension. The parameter set Θ_{sa} of the self-attention layer contains \mathbf{W}_a .

The neighbor user's representation vector is encoded as follows through using max pooling. Maximum pooling implies that the strongest of these features is retained.

$$\mathbf{q}_u^j = [\max(\mathbf{c}^1), \dots, \max(\mathbf{c}^n)], \quad (8)$$

where \mathbf{q}_u^j ($1 \leq j \leq M$) is the embedding for the j -th neighbor user and M is the number of neighbor users. The user's long-term preference \mathbf{p}_u^j and the item representation \mathbf{q}_i are extracted using the same procedure.

3.5. Hierarchical Attention Net. Hierarchical attention mechanism is used to obtain the relationship between neighbor users. Because the relationship between neighbor users is complex, hierarchical modelling is needed to obtain the relationships between different granularities. Intuitively, the order of neighbor user is pivotal for the sequential recommendation. Given a user u and his neighbor users arranged in order, each impact on the item is related to the order. Accordingly, we encode this order information by including position embedding to the user representation: $\mathbf{q}_u^j = \mathbf{q}_u^j + \mathbf{p}_u^j$ ($\mathbf{q}_u = (\mathbf{q}_u^1, \dots, \mathbf{q}_u^{|N_u|})$), where \mathbf{p}_u^j is the embedding for the j -th position.

We use the item representation \mathbf{q}_i to transform the neighbor user representation $\mathbf{q}_u = (\mathbf{q}_u^1, \dots, \mathbf{q}_u^{|N_u|})$ into a user-item representation $\mathbf{p}_u^{s_1}$ as follows:

$$\mathbf{p}_u^{s_1} = \sum_i w_u^i \mathbf{q}_u^i, \quad (9)$$

where $w_u^i \in [0, 1]$ is the weight of \mathbf{q}_u^i . The weighting function w_u^i is computed using the following equation at hop t :

$$\mathbf{o}_u^i = \tanh(\mathbf{W}_{q_u} \mathbf{q}_u^i + \mathbf{W}_p \mathbf{p}_u^{s_1} + \mathbf{W}_{q_i} \mathbf{q}_i), \quad (10)$$

$$w_u^i = \frac{\exp(\mathbf{o}_u^i)}{\sum_i \exp(\mathbf{o}_u^i)}. \quad (11)$$

Note that $\mathbf{p}_u^{s_1}$ is used iteratively as a query. We define the initial value $\mathbf{p}_u^{s_1} = 1/M \sum_{i=1}^{|N_u|} \mathbf{q}_u^i$ at the first hop. Here, $\mathbf{p}_u^{s_1}$ is the sequential preference representation of neighbor users at the union level. That is, $\mathbf{p}_u^{s_1}$ is jointly encoded by all neighbor users. The parameter set Θ_{ha} of the self-attention layer contains \mathbf{p}_u^i , \mathbf{W}_p , \mathbf{W}_{q_u} , and \mathbf{W}_{q_i} .

3.6. Review Pointer. Intuitively, group-buying behavior does not represent individuality. Therefore, in addition to the union-level neighbor user representation, it also needs the individual-level neighbor user representation. For example, a user will buy a Samsung mobile phone just because most neighbor users bought a Samsung mobile phone. In this case, purchases other than Samsung are just noises, which are considered by the recommendation model. We can simply consider $\mathbf{p}_u^{s_1}$ as the final short-term preference derived from user u , but it only captures sequential patterns at the union level. To better express users' short-term preferences, we further explore the influence of neighbor users on purchase action at an individual level, that is, identify several users associated with the item.

Inspired by the pointer mechanism for review-based recommendation [4], we choose the item with the maximum attention weight as follows:

$$i = \arg \max_m (w_u^m), \quad (12)$$

$$\mathbf{p}_u^{s_2} = \mathbf{q}_u^i. \quad (13)$$

3.7. User Representation Vector. The two short-term preference representations of the user must be balanced. We utilize the attention mechanism of dot product for a weighted average of union-level and individual-level short-term preferences. The final short-term user preference is calculated as follows:

$$\mathbf{p}_u^s = \beta_u^1 \mathbf{p}_u^{s_1} + \beta_u^2 \mathbf{p}_u^{s_2}, \quad (14)$$

where β_u^1 and β_u^2 are the weights of union-level short-term preference $\mathbf{p}_u^{s_1}$ and individual-level short-term preference $\mathbf{p}_u^{s_2}$, respectively. The parameter set Θ_{an} of the attention layer contains \mathbf{W}_n , which is the matrix required to calculate attention.

The user's long-term and short-term preferences also need to be balanced. The user's current preference is formed through a linear combination of the representation vector as follows:

$$\mathbf{p}_u = (1 - \alpha) \mathbf{p}_u^l + \alpha \mathbf{p}_u^s, \quad (15)$$

where α balances the importance of the two components and is the hyperparameter. The structure of the item representation network is similar to that of the user representation network; however, it only uses the nearest neighbor items. The corresponding parameters are different.

4. Model Training

We use a factorization machine [29] to obtain the first-order and second-order features. As in collaborative filtering or matrix factorization techniques, the user and item representations are mapped into the same vector representation space \mathcal{R}^n . The rating \hat{r}_{ui} can be computed using the inner product $\mathbf{p}_u^T \mathbf{p}_i$ [2, 42]. \hat{r}_{ui} is a scalar, representing the user-item interaction.

$$\hat{r}_{ui} = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j. \quad (16)$$

The concatenation x of the user representation \mathbf{p}_u and item representation \mathbf{q}_i is passed into a factorization machine. Owing to the symmetry of the scalar product and the inner product, subscript j takes a larger value than i .

Training. This paper uses the mean squared error (MSE) as the objective function. All parameters in the component networks are trained jointly through a backpropagation training procedure. The model parameters of RANM include word embeddings, position embeddings, aspect transformation matrices, attention networks, and CNN networks.

Pretraining. We replaced the hierarchical attention layer with two feed-forward networks. One is for current user, and the other is for the interaction part between neighbor users and items. This simplified model does not consider the interaction details between neighbor users and items.

Generalization. Many research studies have found that deep neural networks tend to suffer from overfitting. Thus, we apply a L2 regularization to first-order weights, second-order weights, and biases. In addition to L2 regularization, we also use dropout [13] to further reduce overfitting. Dropout techniques can prevent co-adaptation by dropping some neural units during the training procedure [43].

5. Experiment

5.1. Experimental Setup.

Dataset: we perform our experiments on the Amazon-5cores dataset (<http://jmcauley.ucsd.edu/data/amazon/>). This dataset contains the product purchase histories from Amazon ranging from May 1996-July 2014. We adopt 5-core settings for these datasets. We only retain the records in Yelp in 2018 (<https://www.yelp.com/dataset/challenge/>) as the final dataset, which is denoted as Yelp18. For all the datasets, a standard text preprocessing method is used to process reviews. Sparsity is defined as $\#Rating / (\#User \times \#Item)$.

For each dataset, we randomly build the training and testing sets in a ratio of 80:20. Moreover, 10% of the records in the training set are the development set for cross-validation. Basic statistics information of the datasets is presented in Table 2.

Metrics: here, we use the MSE as a performance metric. A statistical significance test (using * to show the result) is conducted by performing a pairwise Student’s t -test.

5.1.1. Baseline Methods. To analyze our model performance more comprehensively, we use three types of baselines to conduct extensive experiments. This can not only comprehensively analyze the performance of RANM but also compare the performance of these classical baselines. We introduce the following three categories.

- (a) Probabilistic matrix factorization: PMF [44].
- (b) Latent topic and shallow embedding models with reviews: RBLT [45] and CMLE [46].
- (c) Deep learning-based method with reviews including attention mechanism and aspect modelling: DeepCoNN [13], TransNet [1], D-Attn [2], TARMF [3], MPCN [4], ANR [5], and CARP [14].

Hyperparameter settings: We apply a grid search method for hyperparameter optimization. The final performances are reported over 5 runs. The dimension is selected from {50, 100, 200, 300}. The final dimension size of the word embedding is set to 300. The batch sizes for smaller and bigger datasets are set to be 100 and 200, respectively. For the proposed RANM, we set the number of neighbors as $M=5$. The number of aspects is set to $K=5$, and the embedding size is set to $d=100$, $n=10$, and $h=\{1, 3, 5, 7, 9\}$ for the convolution layer. The number of aspects is set as $H=5$. The learning rate of the learning algorithm is set to 0.001 for model training, weighting factor is set as $\alpha=0.1$ for balancing long-term and short-term preferences, and regularization coefficient is set as $\lambda=0.0001$ to overcome overfitting.

5.2. Results and Discussion. The best configuration is used in the experiment for all baseline methods. For the deep learning method, the same TensorFlow implementation platform is used to facilitate a fair comparison. Table 3 shows the results for all 25 open datasets. Among the baselines, there is no dominating winner across all datasets. Different datasets have a certain influence on the baseline method. Overall, on most datasets, aspect-based systems (ANR and CARP) are better than review-based systems (DeepCoNN, D-Attn, TransNet, TARMF, and MPCN), where the best performance is achieved by using an attention mechanism (D-Attn, TARMF, and MPCN). The review-based system is better than the shallow semantic model (RBLT and CMLE). The shallow semantic model is better than PMF. The same conclusions can be drawn from the average performance.

TABLE 2: Statistics of the twenty-five datasets.

Datasets	#Users	#Items	#Ratings	Sparsity
Amazon instant video	5,130	1,685	37,126	0.0042
Apps for Android	87,271	13,209	752,937	0.00065
Automotive	2,928	1,835	20,473	0.0038
Baby	19,445	7,050	160,792	0.0012
Beauty	22,363	12,101	198,502	0.00073
Books	52, 643	91, 599	2,84, 108	0.00062
CDs and vinyl	75,258	64,443	1,097,592	0.00023
Cell phones and accessories	27,879	10,429	194,439	0.00067
Clothing shoes and jewelry	39,387	23,033	278,677	0.00031
Digital music	5,541	3,568	64,706	0.0033
Electronic	192,403	63,001	1,689,188	0.00014
Grocery and gourmet food	14,681	8,713	151,254	0.0012
Health and personal care	38,609	18,534	346,355	0.00048
Home and kitchen	66,519	28,237	551,682	0.00029
Kindle store	68,223	61,934	982,619	0.00042
Movies and TV	123,960	50,052	1,697,533	0.00027
Musical instruments	1,429	900	10,261	0.0079
Office products	4,905	2,420	53,258	0.0045
Patio, lawn, and garden	1,686	962	13,272	0.0081
Pet supplies	19,856	8,510	157,836	0.00093
Sports and outdoors	35,598	18,357	296,337	0.00045
Tools and home improvement	16,638	10,217	134,476	0.00079
Toys and games	19,412	11,924	167,597	0.00072
Video games	24,303	10,672	231,780	0.00089
Yelp18	31,831	40,841	1, 666, 869	0.0012

From all the experimental results in Table 3, the RANM method has outperformed all the baselines and has passed the hypothesis tests, demonstrating the robustness of our model. This is because a user’s context better represents the short-term preferences, while other models only consider long-term preferences and do not model short-term preferences well. Overall, the experimental results demonstrate that RANM is useful in modelling sequential reviews for rating predictions. It is worthwhile to highlight that a significant improvement is gained by RANM on Yelp18 which is the sparsest dataset with the least review information in all datasets (see Table 1). Compared with the recently proposed end-to-end aspect-level models (i.e., ANR and CARP), RANM’s performance improves by 8% and 18% on an average, respectively. Compared with latent topic and shallow embedding models with reviews (i.e., RBLT and CMLE), RANM’s performance improved by 42% and 34% on an average. Compared with the earliest PMF model, RANM achieved an average performance improvement of up to 50%. These results show that the proposed system is sufficiently powerful.

5.3. Ablation Study. We perform an ablation study to analyze the functions of different components. The default of this discussion refers to the complete model with all components, and we compare it with its six variants:

TABLE 3: Performance comparison in terms of MSE.

Dataset	PMF	RBLT	CMLE	DeepCoNN	D-Attn	TransNet	TARMF	MPCN	ANR	CARP	RANM
Amazon	1.367	1.325	1.105	1.192	1.218	1.111	1.285	1.352	1.129	1.158	0.983*
Apps	1.601	1.582	1.562	1.552	1.337	1.625	1.375	1.402	1.412	<u>1.325</u>	1.201*
Automotive	1.325	1.158	1.259	1.426	1.313	1.205	1.024	<u>0.967</u>	1.151	0.978	0.945*
Baby	1.385	1.368	1.258	1.469	1.309	1.421	1.306	1.313	1.258	<u>1.111</u>	1.009*
Beauty	1.455	1.572	1.596	1.578	1.389	1.458	1.398	1.385	1.299	<u>1.255</u>	1.198*
Books	1.185	1.205	1.098	1.125	1.220	1.024	1.052	1.156	<u>0.985</u>	0.989	0.942*
CDs	1.256	1.156	1.203	0.952	1.186	1.001	1.123	1.054	<u>0.926</u>	0.930	0.910*
Cell phones	2.452	2.412	2.045	2.118	1.625	1.983	1.689	<u>1.489</u>	1.675	1.652	1.365*
Clothing	1.528	1.569	1.452	1.425	1.384	1.356	1.386	1.124	1.255	<u>1.023</u>	1.005*
Digital music	1.206	0.870	0.885	1.045	0.925	0.913	0.853	0.989	0.889	<u>0.820</u>	0.789*
Electronics	1.856	1.985	1.825	1.657	1.356	1.658	1.698	1.364	<u>1.156</u>	1.256	1.106*
Grocery	1.489	1.568	1.425	1.445	1.395	1.352	1.258	1.135	1.256	<u>1.105</u>	1.004*
Health	1.725	1.689	1.652	1.562	1.336	1.425	1.325	<u>1.245</u>	1.348	1.289	1.204*
Home	1.852	1.762	1.714	1.528	1.425	1.475	1.468	<u>1.235</u>	1.326	1.302	1.210*
Kindle store	1.333	1.258	1.098	0.917	0.986	0.993	0.856	<u>0.786</u>	0.878	0.798	0.724*
Movie	1.526	1.583	1.356	1.288	1.285	1.111	1.200	1.156	1.152	<u>1.017</u>	1.009*
Music	1.398	0.815	0.817	1.175	1.302	0.798	0.943	0.846	1.052	<u>0.773</u>	0.705*
Office	1.092	0.759	0.759	0.729	0.858	1.259	0.789	0.795	1.096	<u>0.719</u>	0.706*
Patio	1.785	1.726	1.689	1.633	1.489	1.521	1.485	<u>1.046</u>	1.354	1.258	1.003*
Pet supplies	1.658	1.628	1.521	1.595	1.446	1.485	1.460	1.385	1.352	<u>1.294</u>	1.213*
Sport	1.428	1.596	1.324	1.301	1.275	1.215	1.115	1.134	1.103	<u>1.059</u>	1.005*
Tools	1.566	0.983	1.020	1.487	1.225	1.003	1.169	1.082	1.256	<u>0.960</u>	0.952*
Toys	1.396	1.425	1.305	1.236	1.106	1.112	1.213	0.985	1.093	<u>1.007</u>	0.920*
Video	1.672	1.143	1.253	1.445	1.346	1.390	1.195	1.301	1.187	<u>1.084</u>	0.985*
Yelp18	1.789	1.987	1.852	1.669	1.458	1.602	1.582	1.469	1.556	<u>1.413</u>	1.398*
Average	1.533	1.445	1.363	1.382	1.288	1.3	1.25	1.159	1.206	<u>1.103</u>	1.02*

The best and second-best results are highlighted in bold and underlined, respectively. A statistical significance test (using * to show the result) is conducted by performing a pairwise Student's t -test.

No shared projection (SP) and aspect-level projection (AP): rather than having aspect-specific transformation matrices, we constrain the model using only a single transformation matrix that is shared across all aspects of extractions. Furthermore, we remove the AP. Instead of modelling the aspect-level CNN, we do not differentiate between the aspects of each word by using a simple CNN.

No pretraining (PT): we forgo the pretraining phase (described in Section 4) for the parameters Θ_{ha} of hierarchical attention, that is, the set of parameters for the hierarchical attention layer.

No self-attention (SA): the self-attention layer is removed directly, and the global feature representation between word vectors is obtained without a self-attention mechanism.

No position embedding (PE) and sequential model (SM): on the one hand, the order of neighbors should have some impact on model performance from intuition. On the other hand, we do not use neighbor users to model the interaction with the target item. Our model can alleviate the problem of data sparsity or cold start to some extent because the SM contains rich aspect-level semantic information about long-term and short-term user preferences.

The results of the ablation study for the representative Toys, Video, Beauty, and Book datasets are shown in Table 4. According to the experiment, the removal of neighbor users

results in the lowest performance, which demonstrates the importance of sequential modelling. We observe that a lack of shared AP leads to a small performance degradation. This is the smallest performance degradation of all, showing that the information captured by the shared method is not significantly different from that captured by the nonshared method. If the aspect-level project is further removed, the performance will continue to decline to the second-worst one. This shows the importance of aspect-level reviews and sequence modelling of neighbor users. It also demonstrates the effectiveness of our method in the paper.

Other factors also have a certain impact on the results. Relatively speaking, the impact is not significant. For the hierarchical attention layer, we find that the PT procedure provides a better starting optimization point for the entire optimization objective function. For the SA layer, the CNN only obtains local relevance, but not the global relevance. The SA mechanism is just a good supplement to the CNN. The no PE result suggests that temporal information in the form of a neighbor's proximity is a useful signal for a sequential recommendation.

5.4. Parameter Sensitivity.

Effects of #aspect: Figure 2(a) shows the effect of varying the number of aspects from 1 to 10. We observe that a good performance can be obtained using approximately 5 to 7 aspects. The optimal number of aspects in each dataset is slightly different because the

TABLE 4: Comparison of the model variants for the Toys, Video, Beauty, and Book datasets.

Architecture	Toys	Video	Beauty	Book
No SP	0.934	1.023	1.201	0.988
No AP and no SP	<u>1.035</u>	<u>1.362</u>	<u>1.385</u>	<u>1.156</u>
No PT	0.956	1.122	1.212	1.026
No SA	0.962	1.343	1.320	1.089
No PE	1.023	1.025	1.271	1.088
No SM	1.123	1.401	1.482	1.293
Default	0.920	0.985	1.198	0.942

The worst and second-worst results are highlighted in bold and underlined, respectively.

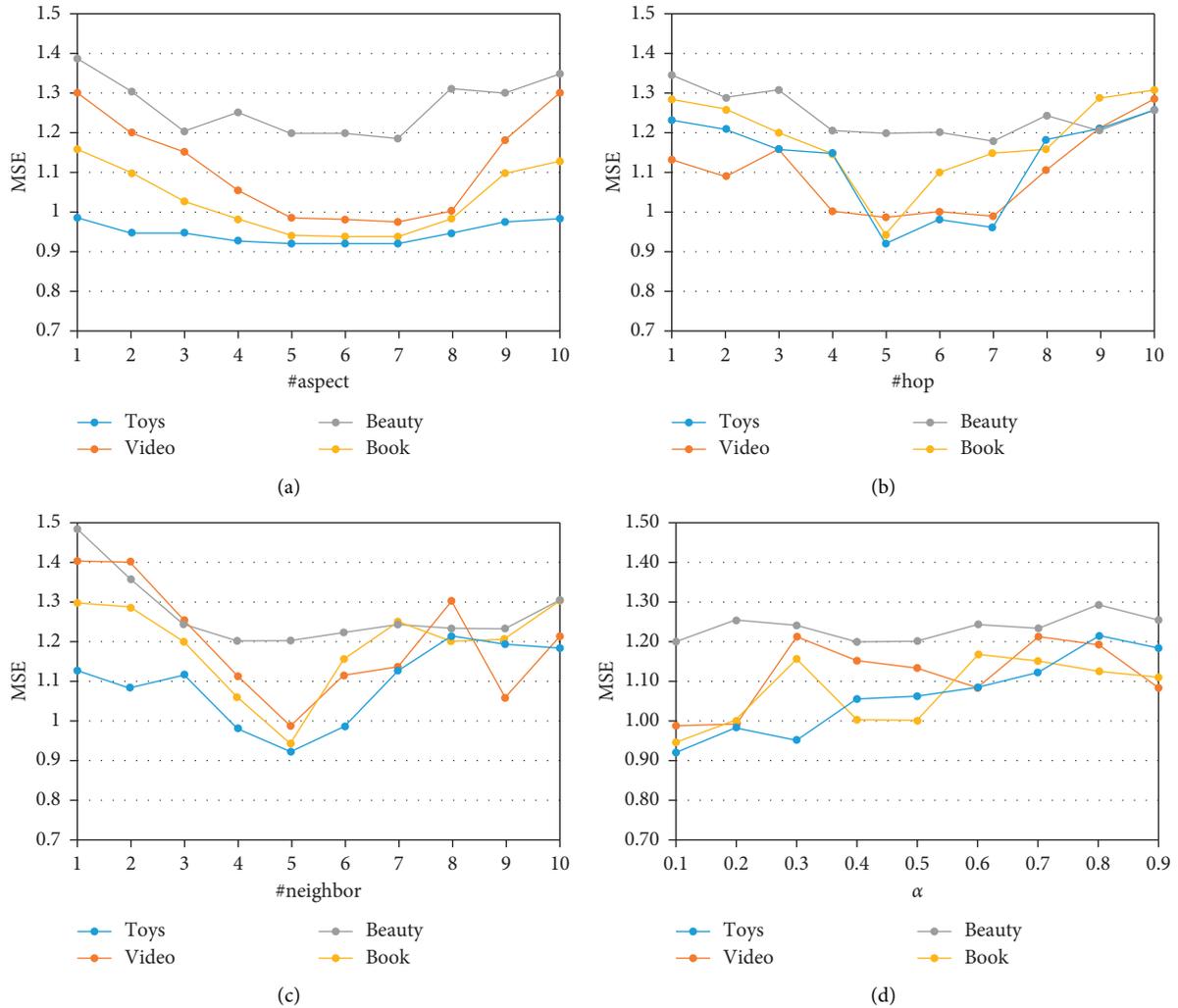


FIGURE 2: Parameter sensitivity of #aspect, #hop, #neighbor, and weighting parameter. (a) Effects of #aspect. (b) Effects of #hop. (c) Effects of #neighbor. (d) Effects of weighting parameter.

number of aspects in the reviews of different datasets is not the same.

Effects of #hop: we show the influence of hops of the hierarchical attention layer in Figure 2(b). Note that the model would only consider the general neighbor user's reviews when the number of hops $H=0$. Figure 2(b) shows the effect of varying the number of hops from 1 to 10 for our model in the test datasets. This indicates

that multiple hops can capture more abstract information from neighbor users' reviews as an external memory. However, too many hops lead to overfitting.

Effects of #neighbor: RANM consistently achieves a better performance at $M=5$, as shown in Figure 2(c). When M becomes smaller or larger, the RANM performance degrades to some extent. This is reasonable because a small M produces less useful neighbor user

TABLE 5: Case study of two user-item pairs from office products.

<i>User1-item1 (ink cartridge)</i>	
User review	The HP 940XL Yellow ink cartridge came in the original sealed box & sealed wrapper. I inserted it into the printer & it has worked great.
Neighbor user review1	This was the least expensive way we could find to purchase manufacturer produced ink cartridges. It works very nicely. I just wish printer ink, in general, wasn't quite so expensive...
Neighbor user review2	Easy to install in Hewlett Packard 6500 Officejet printer, vivid color and accurate printing on HP multipurpose paper.
<i>User2-item2 (folder)</i>	
User review	These folders work great for my filing needs. Assorted colors of folders allow for categorization of contents. The ready tabs are easy to use with a felt tip pen, pencil or ball point.
Neighbor user review1	I like the assorted colors and love the lift up tabs that are quick and easy. However, the weight is fairly standard and not as heavy duty as I would like...
Neighbor user review2	I love these ready tab folders. So far these have been my favorite out of the hanging folders! Love them.

information and a large M would inevitably introduce much noise. Hence, we set $M=5$ in our experiments.

Effects of weighting parameter: parameter α in equation (15) controls the importance of sequential patterns (i.e., union-level and individual-level short-term preferences). Figure 2(d) plots a performance curve with varying α values. When $\alpha=0$, RANM degrades to a recommendation system based on reviews without sequential modelling. We observe that RANM performs much worse in this setting. An optimal performance is consistently achieved when $\alpha=0.1$ in all the three datasets. As α becomes increasingly larger, the performance of RANM further degrades to a larger extent. These results show that short-term user preferences are a useful supplement to long-term user preference. Accordingly, we set $\alpha=0.1$ in our experiments.

5.5. Model Interpretability. Recall that RANM encodes aspect information in the reviews. To better visualize an aspect-level review, we show the top- n phrases whose weight is the sum of the weights of the constituent words. Important phrases are highlighted in red. The weight of words in neighbor users' reviews is obtained by the weight product of equations (7) and (9). The weight of the words in the user's reviews is obtained by equation (7). We randomly sample two users and their neighbor users from office products in the Amazon dataset, as shown in Table 5. From the following two examples, we can see that the RANM model has a more fine-grained representation of users' reviews. Most of the attention to users' and neighbor users' reviews is focused on aspect and opinion words.

6. Conclusions

In this paper, we propose RANM, a novel neural recommendation model for sequential recommendation with reviews. Our model incorporates both user's long-term intrinsic preference and short-term preference to predict the user's rating of the target item. It utilizes aspect-level projections to extract aspect-level sentiment representations. The CNN and the SA layer cooperate with each other to

extract global and local user features (including users to be predicted and neighbor users) and item features from the relevant review documents. A novel hierarchical attention model is proposed to capture the fine-grained alignment probability. At the same time, it uses the attention model to obtain union-level and individual-level short-term preferences. The experimental results show that RANM significantly outperforms various strong state-of-the-art methods. Experiments demonstrate the superiority of our model, especially for sparse data. Furthermore, our model can interpret the recommendation results based on reviews. In future, we will extend RANM to consider the effect of each review on the recommendation. At the same time, the user's interest of point on the item can also be modelled in a more fine-grained and adaptive learning manner.

Data Availability

We perform our experiments on the Amazon-5cores dataset (<http://jmcauley.ucsd.edu/data/amazon/>). This dataset contains product purchase history from Amazon ranging from May 1996 to July 2014. We adopt 5-core settings over these datasets. We only retain the records in Yelp in 2018 (<https://www.yelp.com/dataset/challenge/>) as the final dataset, denoted as Yelp18.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] R. Catherine and W. W. Cohen, *TransNets: Learning to Transform for Recommendation*, RecSys, London, UK, 2007.
- [2] S. Seo, J. Huang, H. Yang, and Y. Liu, *Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction*, RecSys, London, UK, 2017.
- [3] X. Chen, Z. Qin, Y. Zhang, and T. Xu, "Learning to rank features for recommendation over multiple categories," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 305–314, Pisa, Italy, July 2016.

- [4] Y. Tay, A. T. Luu, and S. C. Hui, "Multi-pointer co-attention networks for recommendation," 2018, <https://arxiv.org/abs/1801.09251>.
- [5] J. Y. Chin, K. Zhao, S. Joty, and G. Cong, "ANR: aspect-based neural recommender," in *Proceedings of the 27th ACM International Conference*, pp. 147–156, Turin, Italy, October 2018.
- [6] M. Gaeta, F. Orciuoli, L. Rarità, and S. Tomasiello, "Fitted Q-iteration and functional networks for ubiquitous recommender systems," *Soft Computing*, vol. 21, no. 23, pp. 7067–7075, 2017.
- [7] G. Marques, A. Respício, and A. P. Afonso, "A mobile recommendation system supporting group collaborative decision making," in *Proceedings of the 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems*, vol. 96, pp. 560–567, New York, NY, USA, September 2016.
- [8] M. Hartanto and D. N. Utama, "Intelligent decision support model for recommending restaurant," *Cogent Engineering*, vol. 7, no. No. 1, Article ID 1763888, 2020.
- [9] Y. Kim, "Convolutional neural networks for sentence classification," 2014, <https://arxiv.org/abs/1408.5882>.
- [10] T. Mikolov, M. Karafi'at, L. Burget, J. Cernock'y, and S. Khudanpur, "Recurrent neural network-based language model," in *Proceedings of INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication*, pp. 1045–1048, Chiba, Japan, September 2010.
- [11] L. Wu, C. Quan, C. Li, and D. Ji, "PARL: let strangers speak out what you like," in *Proceedings of the 27th ACM International Conference*, pp. 147–686, Turin, Italy, October 2018.
- [12] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, and X. Luo, "A context-aware user-item representation learning for item recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 2, pp. 1–29, 2019.
- [13] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," 2017, <https://arxiv.org/abs/1701.04783>.
- [14] C. Li, C. Quan, L. Peng et al., "A capsule network for recommendation and explaining what you like and dislike," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 275–284, Paris, France, July 2019.
- [15] T. Hofmann, "Probabilistic latent semantic indexing," pp. 50–57, SIGIR, Columbia, SC, USA, 1999.
- [16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [17] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 448–456, San Diego, CA, USA, August 2011.
- [18] Y. Bao, H. Fang, and J. Zhang, *Topicmf: Simultaneously Exploiting Ratings and Reviews for Recommendation*, AAAI, Palo Alto, CA, USA, 2014.
- [19] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review test," in *Proceedings of the 7th ACM conference on Recommender systems*, vol. 8, Hong Kong, China, February 2013.
- [20] Y. Xu, W. Lam, and T. Lin, "Collaborative filtering incorporating review text and co-clusters of hidden user communities and item groups," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 251–260, Shanghai, China, November 2014.
- [21] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: a rating regression approach," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, NJ, USA, July 2010.
- [22] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, August 2014.
- [23] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinkal, and M. Kankanhalli, "MMALFM: explainable recommendation by leveraging reviews and images," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, p. 16, 2019.
- [24] X. Shao, G. Tang, and B. K. Bao, "Personalized travel recommendation based on sentiment-aware multimodal topic model," *IEEE Access*, vol. 7, pp. 113043–113052, 2019.
- [25] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 83–92, Gold Coast, Australia, September 2014.
- [26] K. Bauman, B. Liu, and A. Tuzhilin, "Aspect based recommendations: recommending items with the most valuable aspects based on user reviews," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 717–725, Halifax, Canada, August 2017.
- [27] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: rating prediction with ratings and reviews," in *Proceedings of the 2018 World Wide Web Conference, International World Wide Web Conferences Steering Committee*, pp. 639–648, Hong Kong, China, May 2018.
- [28] C. Li, X. Niu, X. Luo, Z. Chen, and C. Quan, "A review-driven neural model for sequential recommendation," pp. 2866–3287, 2019, <https://arxiv.org/abs/1907.00590>.
- [29] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web-WWW '10*, pp. 811–820, Raleigh, NC, USA, April 2010.
- [30] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 403–412, Santiago, Chile, August 2015.
- [31] R. He and J. McAuley, "Fusing similarity models with Markov chains for sparse sequential recommendation," in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining*, pp. 191–200, Barcelona, Spain, July 2016.
- [32] X. Chen, H. Xu, Y. Zhang et al., "Sequential recommendation with user memory networks," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 108–116, Los Angeles, CA, USA, February 2018.
- [33] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, <https://arxiv.org/abs/1511.06939>.
- [34] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings*

- of the 1st Workshop on Deep Learning for Recommender Systems, pp. 17–22, Boston, MA, USA, September 2016.
- [35] D. Jannach and M. Ludewig, “When recurrent neural networks meet the neighborhood for session-based recommendation,” in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, Como, Italy, August 2017.
 - [36] M. Ludewig and D. Jannach, “Evaluation of session-based recommendation algorithms,” 2018, <https://arxiv.org/abs/1803.09587>.
 - [37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” 2013, <https://arxiv.org/abs/1310.4546>.
 - [38] J. Pennington, R. Socher, and C. D. Manning, “GloVe: global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, Doha, Qatar, October 2014.
 - [39] C. Ding, T. Li, W. Peng, and H. Park, “Orthogonal non-negative matrix t-factorizations for clustering,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 126–135, Heidelberg, Germany, June 2006.
 - [40] W. Yu, D. Dohan, M.-T. Luong et al., “Combining local convolution with global self-attention for reading comprehension,” 2018, <https://arxiv.org/abs/1804.09541>.
 - [41] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” 2017, <https://arxiv.org/abs/1706.03762>.
 - [42] R. Fänge, U. Lidman, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Secretion of sulfuric acid in *Cassidaria echinophora* Lamarck (Mollusca: mesogastropoda, marine carnivorous snail),” *Comparative Biochemistry and Physiology. A, Comparative Physiology*, vol. 53, no. 1, pp. 101–103, 1976.
 - [43] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
 - [44] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in *NIPS’07: Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 1257–1264, Toronto, Canada, December 2007.
 - [45] Y. Tan, M. Zhang, Y. Liu, and S. Ma, “Rating-boosted latent topics: understanding users and items with ratings and reviews,” in *Proceedings of the 25th international conference on Machine learning-ICML’08*, pp. 2640–2646, Helsinki, Finland, July 2016.
 - [46] W. Zhang, Q. Yuan, J. Han, and J. Wang, “Collaborative multi-level embedding learning from reviews for rating prediction,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2986–2992, New York, NY, USA, July 2016.