Reviews Meet Graphs: Enhancing User and Item Representations for Recommendation with Hierarchical Attentive Graph Neural Network

Chuhan Wu[1], **Fangzhao Wu**[2], **Tao Qi**[1], **Suyu Ge**[1], **Yongfeng Huang**[1],**and Xing Xie**[2]
[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Microsoft Research Asia, Beijing 100080, China
{wu-ch19, qit16, gsy17, yfhuang}@mails.tsinghua.edu.cn,
{fangzwu, xing.xie}@microsoft.com

Abstract

User and item representation learning is critical for recommendation. Many of existing recommendation methods learn representations of users and items based on their ratings and reviews. However, the user-user and item-item relatedness are usually not considered in these methods, which may be insufficient. In this paper, we propose a neural recommendation approach which can utilize useful information from both review content and user-item graphs. Since reviews and graphs have different characteristics, we propose to use a multi-view learning framework to incorporate them as different views. In the review content-view, we propose to use a hierarchical model to first learn sentence representations from words, then learn review representations from sentences, and finally learn user/item representations from reviews. In addition, we propose to incorporate a three-level attention network into this view to select important words, sentences and reviews for learning informative user and item representations. In the graph-view, we propose a hierarchical graph neural network to jointly model the user-item, user-user and item-item relatedness by capturing the first- and second-order interactions between users and items in the user-item graph. In addition, we apply attention mechanism to model the importance of these interactions to learn informative user and item representations. Extensive experiments on four benchmark datasets validate the effectiveness of our approach.

1 Introduction

Precisely learning user and item representations is critical for recommendation (Tay et al., 2018). Many existing recommendation methods learn user and item representations from their rating matrix (Koren et al., 2009; Mnih and Salakhutdinov, 2008; Berg et al., 2017). For example, Koren et al. (2009) proposed to use singular value decomposition (SVD) to learn latent user and item representations based on the review ratings that users gave to items. However, there are massive users and items in online platforms, making the rating matrix between users and items very sparse. Thus, it is difficult for these methods to learn accurate user and item representations (Zheng et al., 2017).

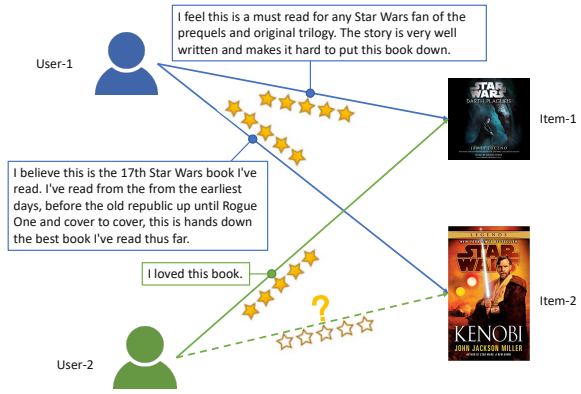


Figure 1: Two example users and items.

In recent years, several deep learning based recommendation methods are proposed to learn user and item representations from review texts (Zheng et al., 2017; Catherine and Cohen, 2017). For example, Zheng et al. (2017) proposed a DeepCoNN approach to learn the representations of users and items from their reviews via convolutional neural networks (CNN). However, these methods only consider the interactions of user-item pairs, while the relatedness between users or items are ignored, which may be insufficient for learning accurate user and item representations.

Our work is motivated by several observations. First, users who write reviews on the same products may have some relatedness. For example, in Fig. 1 both User-1 and User-2 give 5 stars to Item-1, and we can infer that both users are interested in Star Wars. Second, items that commented by the

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4884–4893,
Hong Kong, China, November 3–7, 2019.

same user may also have relatedness. For example, in Fig. 1 both items are commented by User-1, and they share the same topics on Star Wars. Third, modeling the user-user and item-item interactions is useful for recommendation. For example, in Fig. 1 both users may have similar interests on Star Wars, and both books have related topics. Thus, it may be effective to recommend Item-2 to User-2. Fourth, the interactions between users and items, e.g., reviews, may have different importance for representing users and items. For example, in Fig. 1 the interaction between User-1 and Item-1 is more important than that between User-2 and Item-1 in representing this item, since the review of User-1 contain more details about item properties. In addition, different words and sentences in the same review may also have different importance. For instance, the sentence "this is the best book I've read thus far" is more important than "I've read from the earliest days", and the word "best" is more important than "read" in the former sentence.

In this paper, we propose a reviews meet graphs approach (*RMG*) to combine reviews and the information of user-item graphs via a multi-view learning framework. In the review content-view, we use a hierarchical model to learn user and item representations. It first learns sentence representations from words, then learn review representations from sentences, and finally learn user/item representations from reviews. In addition, we propose to apply a three-level attention network to select important words, sentences and reviews to learn informative user and item representations. In the graph-view, we propose to use a graph neural network to capture the user-item, user-user, and item-item relatedness in the user-item bipartite graphs by modeling the first-order and second-order interactions between different users and items. In addition, we propose to incorporate attention mechanism into the graph neural network to model the importance of these interactions for informative user and item representation learning. Extensive experiments on four benchmark datasets validate that our approach can effectively improve the performance of recommendation and outperform many baseline methods.

2 Related Work

Many previous works have studied the problem of learning user and item representations from reviews for recommendation (Zhang et al., 2014; Diao et al., 2014; He et al., 2015; Tan et al., 2016; Ren et al., 2017). Many of existing methods rely on topic models to extract topics from reviews for user and item representation (McAuley and Leskovec, 2013; Ling et al., 2014; Bao et al., 2014). For example, McAuley and Leskovec (2013) proposed a Hidden Factors as Topics (HFT) method to learn latent user and item factors from review content via a latent Dirichlet allocation (LDA) model. Ling et al. (2014) proposed a Ratings Meet Reviews (RMR) approach to represent users and items by first extracting topics from reviews and then aligning the dimensions of these topics with the latent user factors learned from review ratings via matrix factorization. Bao et al. (2014) proposed a TopicMF method to represent users and items by incorporate the topics extracted from review texts to enhance the learning of latent user and item factors from the rating matrix via non-negative matrix factorization (NMF). However, these methods only extract topics from reviews, and cannot effectively utilize the contexts and word orders in reviews, both of which are important for learning accurate user and item representations.

In recent years, several deep learning based methods proposed to learn user and item representations from original review texts (Zhang et al., 2016; Zheng et al., 2017; Catherine and Cohen, 2017; Seo et al., 2017b,a; Chen et al., 2018; Tay et al., 2018; Wu et al., 2019). For example, Zheng et al. (2017) proposed a DeepCoNN approach to learn user and item representations from reviews via CNN networks. Catherine and Cohen (2017) proposed a TransNets approach to use CNNs to learn user and item representations. In their approach, these representations are regularized to be close to the representations of reviews from the target user-item pairs. Seo et al. (2017b) proposed to use CNN networks to learn user and item representations, and applied a word-level attention network to select important words. Chen et al. (2018) proposed to learn review representations using CNNs and they modeled the usefulness of reviews via a review-level attention network to learn informative user and item representation learning. However, these methods can only model the user-item interactions, while the user-user and item-item relatedness is not considered. In addition, these methods usually aggregate reviews or

their texts together, and cannot fully model their informativeness. Different from these methods, in our approach we propose a review meet graph approach which can combine reviews with user-item graphs via a multi-view learning framework to enhance user and item representation learning. In the review text-view, we use a hierarchical framework to learn sentence representations from words first, then learn review representations form sentences, and finally learns user/item representations from their reviews. In addition, we apply attention network at each level to select important words, sentences and reviews. In the graph-view, our approach uses a hierarchical graph neural network to mine the user-item, user-user and item-item relatedness from user-item graphs, and attentively selects the interactions between users and items. Experiments on four benchmark datasets validate the effectiveness of our approach in recommendation.

3 Our Approach

In this section, we will introduce our reviews meet graphs approach for recommendation (denoted as *RMG*). The architecture of our basic *RMG* approach is shown in Fig. 2. Our approach consists of two different views, i.e., a *review content-view* and a *graph-view*.

3.1 Review Content View

The *review content-view* module is used to learn representations of users and items from their review texts. It contains three modules, i.e, *sentence encoder*, *review encoder* and *user/item encoder*.

There are three layers in the *sentence encoder*. The first one is word embedding. It is used to convert a sentence s into a low-dimensional semantic vector sequence. Denote the word sequence of s as $[w_1, w_2, ..., w_M]$, where M is its length. It is converted into a vector sequence $[\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_M]$ via a pre-trained embedding matrix.

The second one is a convolutional neural network (CNN). Local contexts are important for review representation learning. For example, in the sentence "I am a star wars fan", the local contexts of "wars" such as "star" and "fan" are useful for inferring this sentence is about a famous movie series. Thus, we employ a CNN over word embeddings to learn contextual word representations by capturing local contexts. It takes the embedding sequence $[\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_M]$ as input, and outputs a sequence of word contextual representation vectors $[\mathbf{c}_1^w, \mathbf{c}_2^w, ..., \mathbf{c}_M^w]$.

The third layer is a word-level attention network. Different words in the same sentence may have different informativeness in representing users and items. For example, in the sentence "This story book is very interesting" , the word "interesting" is more important than the word "story" in representing this book. Thus, we use an attention network over word representations to learn informative sentence representations by selecting important words. The attention weight of the i-th word w_i is computed as follows:

$$a_i^w = \tanh(\mathbf{w}_w \times \mathbf{c}_i^w + \mathbf{b}_w), \tag{1}$$

$$\alpha_i^w = \frac{\exp(a_i^w)}{\sum_{j=1}^{M} \exp(a_j^w)}, \tag{2}$$

where $\mathbf{w}_w$ and $\mathbf{b}_w$ are parameters. The final representation of the sentence s is the summation of the contextual word representations weighted by their attention weights, i.e., $\mathbf{s} = \sum_{i=1}^{M} \alpha_i^w \mathbf{c}_i^w$.

The *review encoder* module is used to learn representations of reviews from their sentence representations. There are two layers in the *review encoder* module. The first layer is a sentence-level CNN network. Neighboring sentences usually have some relatedness in their content. For example, in the book review "The book is very well written. I will recommend it to others", the two neighboring sentences have close relatedness and they both express positive opinion towards the book. Thus, we use a sentence-level CNN network to learn contextual sentence representations by capturing sentence-level local contexts. We denote the representation sequence of the sentences in a review r as $[\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N]$. The CNN layer takes the representation sequence as input, and outputs a sequence of contextual sentence representations, denoted as $[\mathbf{c}_1^s, \mathbf{c}_2^s, ..., \mathbf{c}_N^s]$.

The second layer is a sentence-level attention network. Different sentences in the same review may have different informativeness for modeling users and items. For example, the sentence "The book tells a good story" is more informative than the sentence "I read this book today" in representing this book. Thus, we use a sentence-level attention network to select important sentences to learn informative representations of reviews for recommendation. The attention weight α_i^s of the i-th sentence is formulated as follows:

$$a_i^s = \tanh(\mathbf{w}_s \times \mathbf{c}_i^s + \mathbf{b}_s), \tag{3}$$

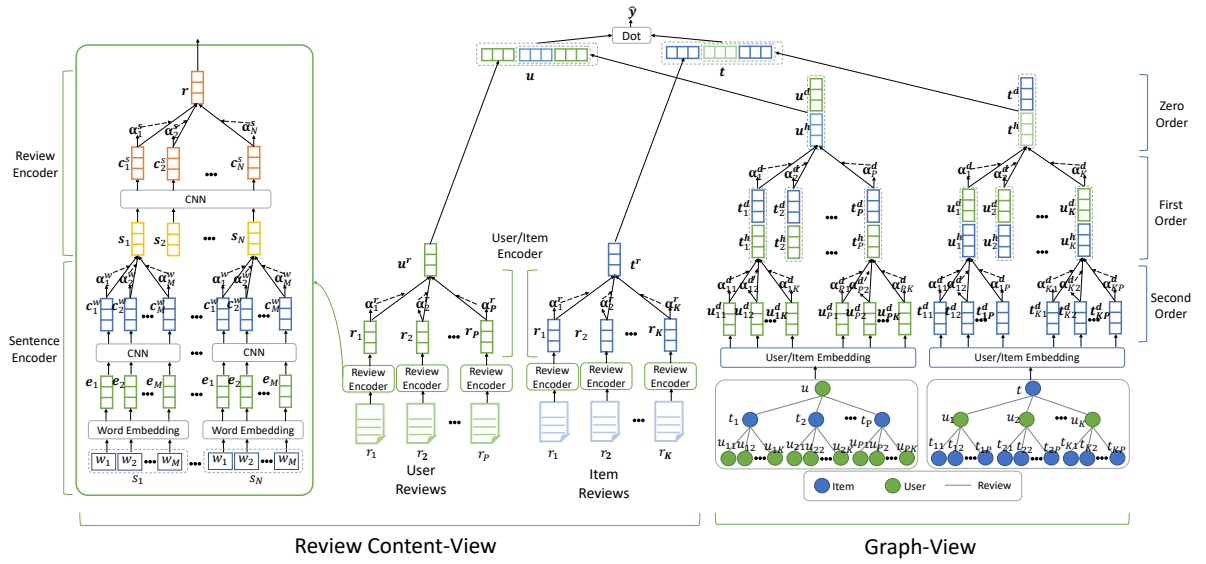


Figure 2: The framework of our *RMG* approach.

$$\alpha_i^s = \frac{\exp(a_i^s)}{\sum_{j=1}^{N} \exp(a_j^s)}, \tag{4}$$

where $\mathbf{w}_s$ and $\mathbf{b}_s$ are parameters in the attention network. The final contextual representation of the review r the summation of the contextual sentence representations weighted by their attention weights, i.e., $\mathbf{r} = \sum_{i=1}^{N} \alpha_i^s \mathbf{c}_i^s$. We apply the *review encoder* to encode the reviews of users and items into their representation vectors.

The *user/item encoder* is used to learn representations of users and items from the their review representations. Different reviews usually have different informativeness in characterizing users and items. For example, in Fig. 1, the first review of Item-1 is more informative than its second review. Thus, we use a review-level attention network to recognize informative reviews and attend to them. Denote the reviews written by a user as $[r_1, r_2, ..., r_P]$, where P is the number of reviews. The attention weight of the review r_i is formulated as follows:

$$a_i^r = \tanh(\mathbf{w}_r \times \mathbf{r}_i + \mathbf{b}_r), \tag{5}$$

$$\alpha_i^r = \frac{\exp(a_i^r)}{\sum_{j=1}^{P} \exp(a_j^r)}, \tag{6}$$

where $\mathbf{w}_r$ and $\mathbf{b}_r$ are parameters. The user representation learned from reviews is the summation of the review representations weighted by their attention weights, i.e., $\mathbf{u}^r = \sum_{i=1}^{P} \alpha_i^r \mathbf{r}_i$. Denote the K reviews commented on an item as $[r_1, r_2, ..., r_K]$. The item representation $\mathbf{t}^r$ learned from these reviews is computed similarly, i.e., $\mathbf{t}^r = \sum_{i=1}^{K} \alpha_i^r \mathbf{r}_i$, where α_i^r is the attention weight of the review r_i.

3.2 User-Item Graph View

The *graph-view* module is used to learn representations of users and items by modeling their interactions in the user-item bipartite graph, as shown in Fig. 2. In this bipartite graph, users and items are represented as nodes, and the interactions of user-item pairs are regarded as edges. However, different from typical graph neural networks (Veličković et al., 2017), the input of typical review-based recommender systems is only a user-item pair (two nodes with an edge) rather than the entire user-item graph. Therefore, it is difficult to directly capture high-order interactions between users and items via a stacked graph neural networks. To solve this problem, in our approach we propose a hierarchical attentive graph neural network for recommendation which can model high-order user-item interactions. It contains three core components, i.e., *second-order* encoder, *first-order* encoder, and *zero-order* encoder. We will introduce each one as follows.

The first one is a *second-order* encoder. Since a user-item graph is bipartite, there are no edges among users (or items). Thus, it is difficult to directly model their relatedness. Luckily, users bought the same items may have some relatedness, as well as the items bought by the same users. Thus, we propose to indirectly mine the user-

user and item-item relatedness by modeling the second-order interactions in the user-item graph, as shown in Fig. 2. For each user-item pair (u, t) in the training set, we denote the items that this user gave ratings to as $[t_1, t_2, ..., t_P]$. Among these items, we denote the users who commented the i-th item as $[u_{i1}, u_{i2}, ..., u_{iK}]$. Motivated by typical graph neural networks (Veličković et al., 2017), we incorporate the embedding of user and item IDs as the node representation in the user-item graph, and then learn the representations of each item from the representations of the users who has commented this item. Usually different users who bought the same item may have different informativeness in representing this item. For example, in Fig. 1, the first user is more important than second user in representing the first item. Thus, we propose to use an attentive graph neural network to model the importance of the users that connected to the item node. Denote the representations of these users as $[\mathbf{u}_{i1}, \mathbf{u}_{i2}, ..., \mathbf{u}_{iK}]$. The attention weight a_{ij}^d of the j-th user who comments the item t_i is computed as:

$$a_{ij}^d = \tanh(\mathbf{w}_2 \times \mathbf{u}_{ij} + b_2), \tag{7}$$

$$\alpha_{ij}^d = \frac{\exp(a_{ij}^d)}{\sum_{k=1}^{K} \exp(a_{ik}^d)}, \tag{8}$$

where $\mathbf{w}_2$ and b_2 are parameters in this network. The user-based representation $\mathbf{t}_i^h$ of the item t_i is the summation of the embeddings of its related user weighted by their attention weights, which is formulated as:

$$\mathbf{t}_i^h = \sum_{k=1}^{K} \alpha_{ik}^d \mathbf{u}_{ik}, \tag{9}$$

The second one is a *first-order* encoder. Since some latent properties of this item may not be revealed by the interactions with users, we incorporate the embedding of item IDs to enhance representation learning. The final representation of node t_i is the concatenation of the item node representation $\mathbf{t}_i^h$ learned from user node representations and the item ID embedding $\mathbf{t}_i^d$, i.e., $\mathbf{t}_i = [\mathbf{t}_i^h, \mathbf{t}_i^d]$. The representations of the users (denoted as $[\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_K]$) who interact with the item t are computed similarly. Similar with the *second-order* encoder, we first learn the item-based representation $\mathbf{u}^h$ of the user u from the representations of its neighbor item nodes in an attentive manner. Denote the attention weight of the i-th item that the user u commented as a_i^d, which is computed as:

$$a_i^d = \tanh(\mathbf{w}_1 \times \mathbf{t}_i + b_1), \tag{10}$$

$$\alpha_i^d = \frac{\exp(a_i^d)}{\sum_{p=1}^{P} \exp(a_p^d)}, \tag{11}$$

where $\mathbf{w}_1$ and b_1 are parameters. The item-based representation $\mathbf{u}^h$ of the user u is the summation of the item representations associated with this user weighted by their attention weights, which is formulated as:

$$\mathbf{u}^h = \sum_{p=1}^{P} \alpha_p^d \mathbf{t}_p, \tag{12}$$

The third one is *zero-order* encoder. It learns the graph-based representation of user u by using the concatenation of the user node representation $\mathbf{u}^h$ and the user ID embedding $\mathbf{u}^d$, i.e., $\mathbf{u}^g = [\mathbf{u}^h, \mathbf{u}^d]$. The representation of the item t is computed similarly. Denote the representation of the item t learned from user-item interactions as $\mathbf{t}^h$, and the embedding of its ID as $\mathbf{t}^d$. Then its graph-based representation $\mathbf{t}^g$ is calculated as $\mathbf{t}^g = [\mathbf{t}^h, \mathbf{t}^d]$.

3.3 Rating Scoring

In our approach, the final representations of users and items are the concatenation of the representations learned from the review content-view and graph-view, i.e., $\mathbf{u} = [\mathbf{u}^g, \mathbf{u}^r]$ and $\mathbf{t} = [\mathbf{t}^g, \mathbf{t}^r]$. The rating score of a user-item pair is predicted by the inner product of user and item representations, i.e., $\hat{y} = \mathbf{u}^T \mathbf{t}$.

3.4 Model Training

For model training, mean squared error is used as the loss function:

$$\mathcal{L} = \frac{1}{T} \sum_{i=1}^{T} (\hat{y}_i - y_i)^2, \tag{13}$$

where T denotes the number of user-item pairs for training, $\hat{y}_i$ and y_i respectively denote the predicted and gold rating score of the i-th user-item pair. By optimizing the loss function, all parameters can be tuned via backward propagation.

4 Experiments

4.1 Datasets and Experimental Settings

We conducted experiments on four widely used benchmark datasets in different domains and

Dataset	#users	#items	#reviews
Toys	19,412	11,924	167,597
Kindle	68,223	61,935	982,619
Movies	123,960	50,052	1,679,533
Yelp	199,445	119,441	3,072,129

Table 1: Statistics of the benchmark datasets.

scales to validate the effectiveness of our approach. Following (Chen et al., 2018), we used three datasets from the Amazon collection[1](He and McAuley, 2016), i.e., *Toys_and_Games*, *Kindle_Store*, and *Movies_and_TV* (respectively denoted as *Toys*, *Kindle* and *Movies*). Another dataset is from Yelp Challenge 2017[2] (denoted as **Yelp**), which is a large collection of restaurant reviews. Following (Chen et al., 2018), we only kept the users and items which have at least 5 reviews. The detailed statistics of the four datasets are summarized in Table 1. The ratings in these datasets are ranged in [1, 5].

In our experiments, the word embeddings were 300-dimensional. We used the pre-trained Google embedding (Mikolov et al., 2013) for initialization. The CNNs had 150 filters and their window sizes were set to 3. The user and item embedding vector was 100-dimensional. We applied 25% dropout (Srivastava et al., 2014) to each layer in our model to mitigate overfitting. Adam (Kingma and Ba, 2014) was used as the optimization algorithm. The batch size was set to 20. We randomly selected 80% of the user-item pairs in each dataset for training, 10% for validation and 10% for test. All hyperparameters were tuned according to the validation set. We independently repeated each experiment 5 times and reported the average Root Mean Square Error (RMSE).

4.2 Performance Evaluation

We evaluate the performance of our approach by comparing it with several baseline methods[3]. The methods to be compared include:

- *PMF*: Probabilistic Matrix Factorization, which models users and items based on ratings via matrix factorization (Mnih and Salakhutdinov, 2008).
- *NMF*: Non-negative Matrix Factorization for recommendation based on rating scores (Lee and Seung, 2001).
- *SVD++*: The recommendation method based on rating matrix via SVD and similarities between items (Koren, 2008).
- *HFT*: Hidden Factor as Topic (HFT), a method to combine reviews with ratings via LDA (McAuley and Leskovec, 2013).
- *DeepCoNN*: Deep Cooperative Neural Networks, a neural method to jointly model users and items from their reviews via CNN (Zheng et al., 2017).
- *Attn+CNN*: Attention-based CNN, which uses both CNN and attention over word embeddings to learn user and item representation from reviews (Seo et al., 2017b).
- *NARRE*: Neural Attentional Rating Regression with Review-level Explanations, which uses attention mechanism to model the informativeness of reviews for recommendation (Chen et al., 2018).
- *RMG-review*: A variant of our approach with the review content-view only.
- *RMG*: Our reviews meet graphs approach.

In Table 2, we present a simple comparison of different methods by summarizing the information they incorporated. Traditional matrix factorization methods such as *PMF*, *NMF* and *SVD++* learn user and item representations based on review ratings only, while other methods can exploit both rating scores and reviews texts to learn representations for recommendation. Among these methods, *HFT* is based on topic models and cannot effectively utilize the contexts and word orders of words. *DeepCoNN* and *Attn+CNN* simply concatenate all reviews into a long document, and cannot model the informativeness of different reviews. Although *NARRE* can model review informativeness using a review-level attention network, it simply aggregates all sentences and cannot model the informativeness of words and sentences. *RMG-review* can simultaneously recognize important word, sentences and reviews, but does not consider the information in user-item graphs. Different from these methods, our *RMG*

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

[2] https://www.yelp.com/dataset_challenge

[3] Since in our framework the input is a single user-item pair, other popular graph neural architectures such as GCN may not be applicable. We will also explore incorporating them in our future work.

Information	PMF	NMF	SVD++	HFT	DeepCoNN	Attn+CNN	NARRE	RMG-review	RMG
Rating score	✓	✓	✓	✓	✓	✓	✓	✓	✓
Review text				✓	✓	✓	✓	✓	✓
Context & word order					✓	✓	✓	✓	✓
Review attention							✓	✓	✓
Word attention						✓		✓	✓
Sentence attention								✓	✓
User-item graph									✓

Table 2: Comparisons of the information used in different methods.

Methods	Toys	Kindle	Movies	Yelp
PMF	1.3076	0.9914	1.2920	1.3340
NMF	1.0399	0.9023	1.1125	1.2916
SVD++	0.8860	0.7928	1.0447	1.1735
HFT	0.8925	0.7917	1.0291	1.1699
DeepCoNN	0.8890	0.7876	1.0128	1.1642
Attn+CNN	0.8805	0.7796	0.9984	1.1588
NARRE	0.8769	0.7783	0.9965	1.1559
RMG-review	0.8685	0.7511	0.9679	1.1310
RMG	**0.8438**	**0.7366**	**0.9514**	**1.1113**

Table 3: The performance of different methods on the benchmark datasets. The metric is RMSE.

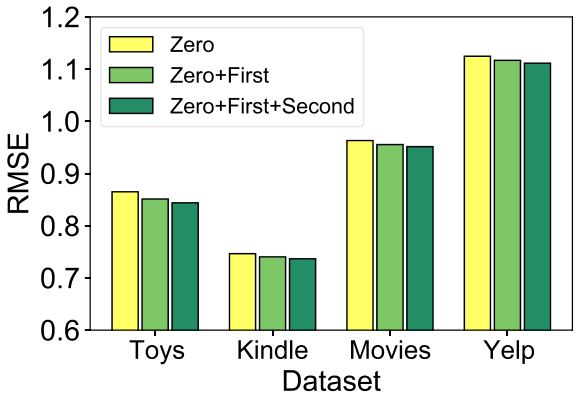


Figure 3: The influence of the depth of the hierarchical attentive graph neural network.

approach can learn user and item representations from both reviews and graphs, which has the potential to learn more informative user and item representations.

The performance of different methods is summarized in Table 3. The results lead to several observations. First, the methods which combine reviews and ratings (e.g., *HFT*, *DeepCoNN*, *NARRE* and *RMG*) usually outperform the methods based on ratings only (*PMF*, *NMF* and *SVD++*). This is probably because reviews usually contain rich clues on user preferences and item properties, which is useful for user and item representation learning for recommendation.

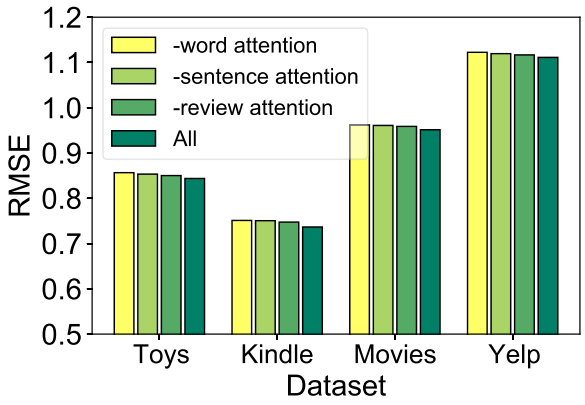
Second, among the methods considering the content information of reviews, neural network based methods (e.g., *DeepCoNN*, *Attn+CNN*, *NARRE* and *RMG*) usually outperform *HFT*, which is based on topic models. This is probably because *HFT* rely on bag-of-words features to extract topics from reviews, and cannot utilize the contexts and orders of words. Different from topic model based methods, neural methods can better model user preferences and item properties by utilizing the rich semantic information in reviews, which is beneficial for recommendation.

Third, among the methods based on deep learning, *RMG-review* and *RMG* can consistently outperform other baseline methods. This is because different words, sentences and reviews usually have different informativeness for modeling users and items. Different from *Attn+CNN*, *NARRE* and *DeepCoNN*, both *RMG-review* and *RMG* use a hierarchical model to learn user and item representations. In addition, our approach can select important words, sentences and reviews via a three-level attention network to help learn more accurate user and item representations for recommendation.

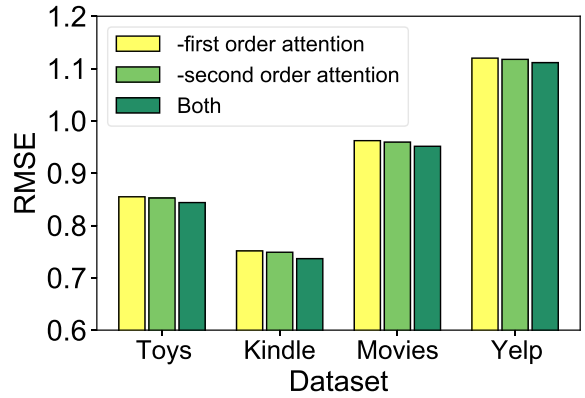
Fourth, our *RMG* approach which incorporates information of user-item graph outperforms other methods based on reviews only (e.g., *RMG-review*), and further hypothesis test results show that the advantage over *RMG-review* is significant. This is because the information of user-item graphs is useful for recommendation, and modeling the interactions between users amd items in graphs can help learn more accurate user and item representations. This result validates the effectiveness of our approach.

4.3 Influence of the Depth of Graph Neural Network

In this section, we conducted several experiments to explore the influence of the depth of the graph neural network on our approach. We compare the performance of our approach with its variants



(a) Attentions in the review content-view.



(b) Attentions in the graph-view.

Figure 4: The effectiveness of different attentions.

with the zero-order encoder only or with both the zero- and first-order encoder. The results are illustrated in Fig. 3. According to the results, we find the performance of our approach can be consistently improved as the depth increases. This is because the first-order information of graph contains the interactions between users and items, and the second-order information can reveal the user-user and item-item relatedness. Thus, more information can be incorporated as the depth increases, which can benefit user and item representation learning. Although the performance may be further improved when the depth gets deeper, in our approach we only set the depth of the graph neural network to 2. This is because the user-item, user-user and item-item interactions are usually sufficient for modeling users and items, and their high-order relatedness is usually weak. In addition, the computational cost is huge when higher-order information is considered. Thus, a moderate depth of the graph neural network (i.e., 2) is the most appropriate for our approach.

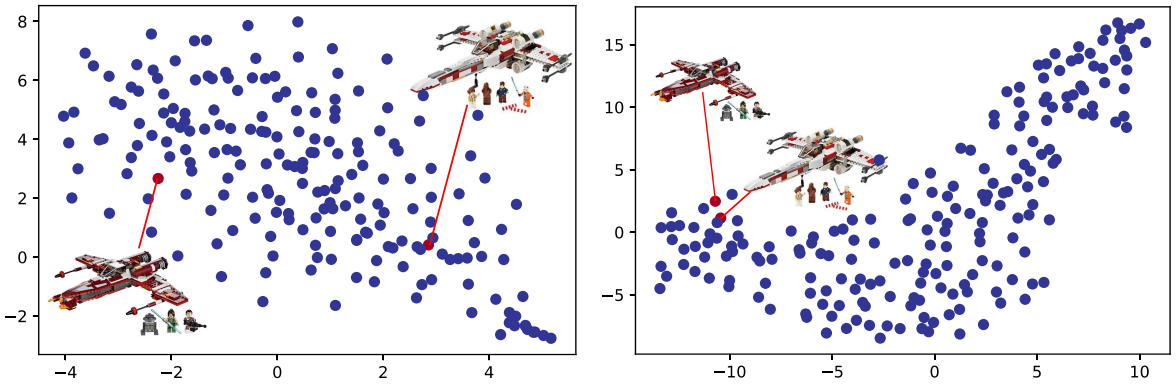
4.4 Effectiveness of Attention Mechanism

In this section, we conducted experiments to explore the effectiveness of the attention networks in our approach. First, we want to validate the effectiveness of the attention network in the review content-view. We compare the performance of several variants of our approach with one kind of attention network removed to evaluate the contribution of different attentions. The results are shown in Fig. 4(a). From Fig. 4(a), the word-level attention can effectively improve the performance of our approach. This is because different words in the same review have different importance in representing this review. Therefore, selecting important words via a word-level attention network can learn more informative sentence representations. In addition, the sentence-level attention is also useful. This is because different sentences also have different informativeness, and selecting important sentences using a sentence-level attention network can benefit review representation learning. Besides, the review-level attention is also useful in our approach. This is because different reviews have different informativeness for learning user and item representations, and distinguishing informative reviews from the uninformative ones can help learn more accurate user and item representations. Moreover, combining all three kinds of attentions can further improve our approach, which validates the effectiveness of our hierarchical attention framework.

Then, we want to validate the effectiveness of the attention network in the graph-view. We compare our approach with its variants by removing the attention networks in the first-order or second-order encoder to validate the effectiveness of them. The results are shown in Fig. 4(b). According to Fig. 4(b), we find the attention networks in both first- and second-order encoders are important in our approach. This may be because the interactions between users and items usually have different importance in representing them, and distinguish informative interactions from uninformative ones is useful for learning better user and item representations. These results validate the effectiveness of the attention networks in the graph-view.

4.5 Case Study

In this section, we conducted several case studies to visually explore our *RMG* approach can learn better user and item representations. For instance,



(a) *RMG-review* (b) *RMG*

Figure 5: Visualization of the item representations learned by *RMG-review* and *RMG* via t-SNE. Each point represents an item. The red points represent two similar items.

we use t-SNE to visualize the representations of several randomly selected items in the *Toys* dataset learned by *RMG-review* and *RMG*. Among these items there are two very similar items, both of which are Lego toys of the planes in Star Wars. The results are respectively shown in Fig. 5(a) and 5(b)[4]. From the results, we find the distance between the representations of these two items learned by *RMG-review* is large, which usually indicates their similarity is low. This may be because the reviews of both items are insufficient (only 3 reviews appeared in the training set), making it difficult for *RMG-review* to learn accurate representations of them. Different from *RMG-review*, our *RMG* approach can correctly recognize that both items are very similar. This is probably because our approach can mine the user-user and item-item relatedness by modeling the first-order and second-order interactions between users and items. Thus, our *RMG* approach can learn better user and item representations.

5 Conclusion and Future Work

In this paper, we propose a reviews meet graphs approach for recommendation which can exploit information of user-item graphs. In our approach, we use a multi-view framework to incorporate review texts and user-item graphs as different views. In the review content-view, we use a hierarchical model to first learn sentence representations from words, then learn review representations from sentences, and finally learn user/item representations from reviews. In addition, we apply a three-level attention network to select important words, sentences and reviews for informative representation learning. In the graph-view, we propose to use a graph neural network to jointly mine user-item, user-user and item-item relatedness by modeling the first- and second-order interactions between users and items in graphs. In addition, we propose to apply attention mechanism to model the importance of these interactions to learn more accurate user and item representations. Experiments on four benchmark datasets validate the effectiveness of our approach.

In our future work, we will explore several potential directions. First, since we only use the local neighbors of nodes in the user-item graph, we will explore the use of other kinds of graph neural networks to further enhance the learning of users and items. Second, since there are also other kinds of user and item information, we will explore how to incorporate heterogeneous user and item information to benefit recommendation. Third, we will work on how to reduce the computational cost of our approach, which can improve the applicability of our approach.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant number 2018YFC1604002, the National Natural Science Foundation of China under Grant numbers U1836204, U1705261, U1636113, U1536201, and U1536207.

[4]We aims to show the relative similarities between the representations of the two items, and their absolute distances are not comparable in the two figures.

References

Yang Bao, Hui Fang, and Jie Zhang. 2014. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *AAAI*, volume 14, pages 2–8.

Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *RecSys*, pages 288–296. ACM.

Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *WWW*, pages 1583–1592.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *KDD*, pages 193–202.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, pages 507–517.

Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *CIKM*, pages 1661–1670.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434.

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

Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562.

Guang Ling, Michael R Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *RecSys*, pages 105–112.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*, pages 165–172.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264.

Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social collaborative viewpoint regression with explainable recommendations. In *WSDM*, pages 485–494.

Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017a. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *RecSys*, pages 297–305. ACM.

Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017b. Representation learning of users and items for review rating prediction using attention-based convolutional neural network. In *MLRec*.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-boosted latent topics: Understanding users and items with ratings and reviews. In *IJCAI*, pages 2640–2646.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-pointer co-attention networks for recommendation. In *KDD*, pages 2309–2318.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Chuhan Wu, Fangzhao Wu, Junxin Liu, and Yongfeng Huang. 2019. Hierarchical user and item representation with three-tier attention for recommendation. In *NAACL*, pages 1818–1826.

Wei Zhang, Quan Yuan, Jiawei Han, and Jianyong Wang. 2016. Collaborative multi-level embedding learning from reviews for rating prediction. In *IJCAI*, pages 2986–2992.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*, pages 83–92.

Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*, pages 425–434. ACM.