A Zero Attentive Relevance Matching Network for Review Modeling in Recommendation System

Hansi Zeng, Zhichao Xu, Qingyao Ai

School of Computing, University of Utah, Salt Lake City, UT, USA, 84112
{hanszeng,brutusxu,aiqy}@cs.utah.edu

Abstract. User and item reviews are valuable for the construction of recommender systems. In general, existing review-based methods for recommendation can be broadly categorized into two groups: the siamese models that build static user and item representations from their reviews respectively, and the interaction-based models that encode user and item dynamically according to the similarity or relationships of their reviews. Although the interaction-based models have more model capacity and fit human purchasing behavior better, several problematic model designs and assumptions of the existing interaction-based models lead to its suboptimal performance compared to existing siamese models. In this paper, we identify three problems of the existing interaction-based recommendation models and propose a couple of solutions as well as a new interaction-based model to incorporate review data for rating prediction. Our model implements a relevance matching model with regularized training losses to discover user relevant information from long item reviews, and it also adapts a zero attention strategy to dynamically balance the item-dependent and item-independent information extracted from user reviews. Empirical experiments and case studies on Amazon Product Benchmark datasets show that our model can extract effective and interpretable user/item representations from their reviews and outperforms multiple types of state-of-the-art review-based recommendation models.

Keywords: Review Modeling, Interaction-based Model, Relevance Matching

1 Introduction

Review text is considered to be valuable for effectively learning user and item representations for recommender systems. Previous studies show that the incorporation of reviews into the optimization of recommender systems can significantly improve the performance of rating prediction by alleviating data sparsity problems with user preferences and item properties expressed in review text [19,18,38,5]. In general, existing review based recommender systems for rating prediction can be roughly categorized into two groups: 1) The siamese models that independently encode static user and item representations from reviews

and use the static representations to predict the rating [38,5]; 2) The interaction-based models that dynamically learn the user and item representations based on their context [29,10]. In particular, the interaction-based models assume that, given different target items, different user reviews might play different roles in determining the utility of the items. For example, when the target item is about an album from the Led Zeppelin, the user review that reflects her interest on Rock & Roll music might be more useful than the rest of her reviews.

Although the interaction-based models have more model capacity and fit the human purchasing behavior better [29], several problematic model designs and assumptions lead to its lower performance than siamese models as shown in recent studies [23]. First, most existing interaction-based models exploit co-attention mechanism [24,33,6,28] to distill textual similarity information between user and item reviews, but such information might be diluted when there is a vast amount of text in user and item reviews. Second, because the number of reviews to profile each user in the training set is limited, it is common that the target item's characteristics are beyond the interest of a user expressed in her limited reviews. Interaction-based models that force the user representations to extract valuable information from user reviews for the target item might introduce irrelevant aspects of the user and cause serious overfitting. Third, existing interaction-based models extract user-item relationships mostly by modeling the textual similarity between user and item reviews. High textual similarities between user and item reviews, however, not necessarily reflect the user's true opinion on the target item. For example, an item review "the taste of cappuccino is really good" might have higher textual similarity with a user review "I really enjoy the taste of beef pho" than with "I am a big coffee fan", but the latter review that reflects the user opinion on coffee could be more informative when predicting her rating on the target item.

Based on these observations, in this paper, we propose a new interaction-based rating prediction model to mitigate the weakness of the existing interaction-based recommendation models. First, we implement a relevance matching model [11] instead of a semantic matching model [33] to search the relevant review from the user to the target item. Our relevance matching model treats each user review as a query to search and extracts relevant information from all the reviews of the target item. It is capable of discovering relevance information from a large amount of review text with thousands or more words. Second, to better capture the semantic relationships instead of the textual similarity between user and item reviews, we use the *ground-truth* review (available in the training stage) written from the user to the target item and the corresponding item reviews as a pair of positive "query-document" to train our relevance matching module and plug it as the auxiliary loss in the training objective, since the *ground-truth* review expresses the user true opinion to the target item. After the relevance matching function is well-trained, other user reviews that have high relevance matching scores to the target item would share similar characteristics to the *ground-truth* review and also reflect the user true interest to the target. Last but not least, when there is not relevant review from the user to the target item,

we exploit a zero-attention network [1] to avoid using irrelevant reviews to build user representations. Our zero-attention network not only builds dynamic user representations when there are high informative reviews from the user to the target item, but also allows the model to degenerate to a siamese model with static user representations when all user reviews are not relevant to the target item. Specifically, separated from the interaction module, we build static user and item embeddings using a multi-layer convolutional self-attention network to extract information hierarchically from words, sentences and reviews. We then construct the final user representations using both the dynamic user representations extracted by the interaction module and the static user embeddings created by the self-attention network. When there is no user review relevant to the target item, the dynamic user representation created by the interaction module with zero attention networks would be downgraded to a zero vector and the final rating prediction of the user-item pair would purely depend on the static user and item embeddings. Empirical experiments and case studies on four datasets from Amazon Product Benchmark show that that our proposed model the Zero Attentive Relevance Matching Network (ZARM) can extract effective and interpretable user/item representations from review data and outperforms multiple types of state-of-the-art review-based recommendation models.

2 Related Works

Review Based Recommendation. Using review text to enhance user and item representations for recommender system has been widely studied in recent years [19,18,37,14,27,22,36,9]. Many works are focus on topic modeling from review text for users and items. For example, the HFT[19] uses LDA-like topic modeling to learn user and item parameters from reviews. The likelihood from the topic distribution is used as a regularization for rating prediction by matrxi factorization(MF). The RMR[18] uses the same LDA-like model on item reviews but fit the ratings using Guassian mixtures but not MF-like models. Recently, with the advance of deep learning, many recommendation models start to combine neural network with review data to learn user and item representations, including DeepCoNN[38], TransNets[4] D-Att[25], NARRE[5], HUITA[31], HANN[7], MPCN[29], AHN[10], HSACN[35]. Despite their differences, existing work using deep learning models for review modeling can be roughly divided into two styles – siamese networks and interaction-based networks. For example, DeepCoNN[38] uses two convolution neural networks to learn user and item representations from reviews statically; NARRE[5] extends CNN with an attention network over review-level to select reviews with more informativeness. In addition, the MPCN[29], a interaction-based network, uses co-attention mechanism to select the most informative and matching reviews from the user and item respectively, then another attention mechanism is applied to learn the fixed dimensional representation, by modeling the word-level interaction on the matched reviews. However, both of these two styles have their own weaknesses. The siamese models lack the dynamic target-dependent modeling and neglect the interaction between the user and target. But the interaction-based models forcely require

the dynamic matching between each user and item, neglecting the fact that not every user exists the informative review to the target. Even the informative review exists, the matching information might be diluted considering thousands of words within tens of review for profiling the user and item.

Interaction Based Text Matching. The review based dynamic user-item modeling is closely related to query-document representation learning in the QA [24] task or premise-hypothesis encoding in the NLI [33,6,28] task that exploit the co-attention mechanism. The co-attention mechanism computes the pairwise similarity between two sequences, builds the pair-wise attention weights, and integrates them with other feautres of the sequences for effective text semantic matching learning. Besides the text semantic matching in the NLP tasks, several works on the IR tasks[11,32,12,2] also utilize the interaction-based approaches for text relevance matching learning. For example, DRMM [11] proposes a pooling pyramid technique that converts the pair-wise similarity matrix into the histogram, and use it as feature for final text matching prediction. Based on the DRMM, K-NRM [32] introduces the kernel-based differential pooling technique that can learn the matching signal in different level. Recent work[21] further investigate using the semantic matching and relevance matching together or alone in the NLP and IR tasks. It finds that using relevance matching alone performs reasonable well in many NLP tasks but the semantic matching is not effective for IR tasks.

Rethinking the Progress of Deep Recommender System. While we have witnessed the rapid advancements of deep learning methodology and its applications on the field of recommender systems, there are worries about the progress we made. [8] investigated the performance of several recent methodologies proposed in top conferences and found most of them can not compete with traditional methods, like BPR or Item-KNN. Furthermore, [23] focused on the usefulness of reviews. He examined several review-based recommendation algorithms and found that applying complex structures to extract semantic information in reviews, not necessarily improve the system's performance. Our goal is to try to tackle these existing problems in this field and propose an interpretable method to effectively utilize the review information.

3 Proposed Method

3.1 Overview

The goal of the proposed model is to predict the rating from the user to the target item based on their review text. The architecture of our model is shown in Figure. 1. Our model contains two parallel encoders that use multi-layer convolution self-attention network to hierarchically encode user and item static representations from their reviews respectively. Besides, the model has a interaction module that encodes the user dynamic representation according to her current interacted item where we first compute the relevant level of each user review to the target item by the relevance matching function. Then the zero-attention network is applied to allow the dynamic user representation degrade to a zero vector when there is no user review relevant to the target, in which case the final user representation

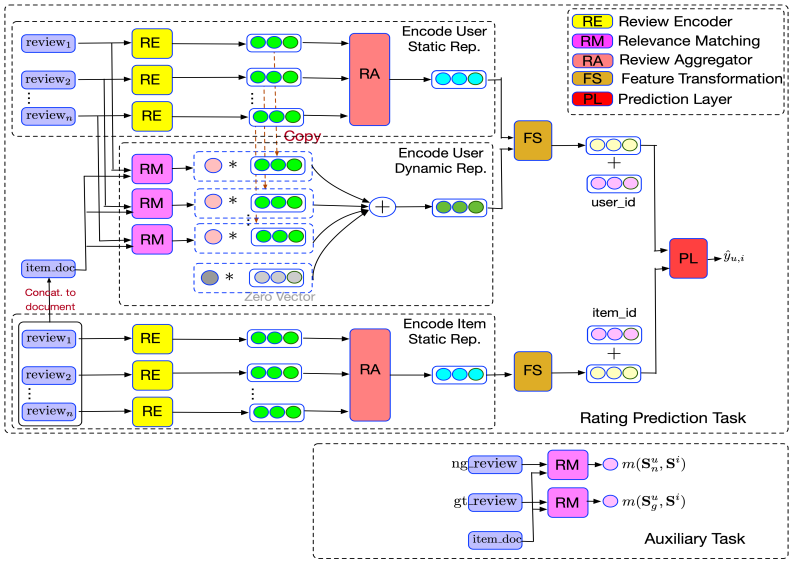

Fig. 1: Overview of our model structure

would purely depend on the static user representation. The encoded user static and dynamic representations will be concatenated and be taken as the input to the feature transformation layer to encode the final user representation. On the rightmost of the model, the prediction layer is added to let the learned user and item final representations interact with each other and compute the final rating prediction. In the training stage, the auxiliary loss is plugged to guide the training of relevance matching function. In the following sections, we will introduce the static user/item encoder (section 3.2), dynamic user encoder composed of the relevance matching function and zero-attention network (section 3.3), prediction layer(section 3.4), and the training objective (section 3.5) in details.

3.2 Static User/Item Encoder

Since the static user and item encoder only differ in their inputs, we introduce the process of encoding user static representation in the following in details. And the same process is applied to static item encoder in the similar way. Assume the input of the user encoder is $\{r_1^u, \dots r_N^u\}$, where N is the number of reviews written by the user. We learn each review representation hierarchically from word-level to sentence-level. More specifically, a user review $r^u = \{s_1, \dots s_T\}$ consists of T sentences, and each sentence s_i is composed of a sequence of L words $\{w_1^i, \dots w_L^i\}$. To learn the sentence representation $\boldsymbol{s_i}$, we apply the word-level self-attentive convolution network to encode the contextual representation of each word in the sentence and use the attention network to aggregate the learned contextual embeddings in to a single vector. Mathematically, we first apply the word embedding layer to map each word w_j^i into a vector $\boldsymbol{w}_j^i \in \mathbb{R}^{d_w}$ to form a sequence of word embeddings $\boldsymbol{W}^i \in \mathbb{R}^{d_w \times L}$, then we apply the word-level convolution neural network to learn the local semantic representation of

each words:

$$\boldsymbol{Q}_w^i = \text{CNN}_w^Q(\boldsymbol{W}^i),\ \boldsymbol{K}_w^i = \text{CNN}_w^K(\boldsymbol{W}^i),\ \boldsymbol{V}_w^i = \text{CNN}_w^V(\boldsymbol{W}^i) \tag{1}$$

where $\boldsymbol{Q}_w^i,\ \boldsymbol{K}_w^i,\ \boldsymbol{V}_w^i \in \mathbb{R}^{d_w \times L}$. To enrich each word semantic representation and capture long-range dependencies between words, we apply the multihead-self-attention network [30] on top of the learned word local representation from CNN$(\cdot)$. Finally, a 1 layer feed-forward network is sequentially plugged to learn more flexible representations:

$$\boldsymbol{Z}_w^i = \text{FFN}_w\Big(\text{Multihead-Self-Attention}_w(\boldsymbol{Q}_w^i, \boldsymbol{K}_w^i, \boldsymbol{V}_w^i)\Big) \tag{2}$$

where $\boldsymbol{Z}_w^i \in \mathbb{R}^{d_s \times L}$. Then we use addictive-attention network [34] to aggregate the contextual representations into a single vector $\boldsymbol{s_i} \in \mathbb{R}^{d_s}$ for sentence modeling:

$$\boldsymbol{s_i} = \text{Addictive-Attention}_w(\boldsymbol{Z}_w^i) \tag{3}$$

We apply the same procedure on each sentence of the review r^u to form a sequence of sentence representations $\boldsymbol{S} = [\boldsymbol{s}_1, \dots \boldsymbol{s}_T] \in \mathbb{R}^{d_s \times T}$. Then we take the sentence sequences as an input to sentence-level self-attentive convolution network with addictive-attention network to form the review representation $\boldsymbol{r}^u \in \mathbb{R}^{d_r}$:

$$\boldsymbol{Z}_s = \text{Multihead-Self-Attention}_s\Big(\text{CNN}_s^Q(\boldsymbol{S}), \text{CNN}_s^K(\boldsymbol{S}), \text{CNN}_s^V(\boldsymbol{S})\Big) \tag{4}$$

$$\boldsymbol{r}^u = \text{Addictive-Attention}_s(\boldsymbol{Z}_s) \tag{5}$$

We apply the same hierarchical network on each review written by the user, then form a sequence of review representations $\boldsymbol{R} = [\boldsymbol{r}_1^u, \dots \boldsymbol{r}_N^u] \in \mathbb{R}^{d_r \times N}$. Finally, we apply the user-level addictive-attention network to aggregate the information of these reviews and form a single vector $\boldsymbol{u}^{static} \in \mathbb{R}^{d_r}$ to form the user static representation:

$$\boldsymbol{u}^{static} = \text{Addictive-Attention}_u(\boldsymbol{R}^u) \tag{6}$$

The item static representation $\boldsymbol{i}^{static}$ can be obtained using a similar procedure.

3.3 Dynamic User Encoder

To learn the dynamic user representation, we first compute the relevant scores of the reviews of the user to the target item using relevance matching function. In other words, given N reviews written by the user $\{r_1^u, \dots r_N^u\}$, we want to compute their corresponding relevance scores $\{\alpha_1, \dots \alpha_N\}$ to the target. The detailed introduction of the relevance matching function is in the following.

Relevance Matching Function: The input of the function is a query-document pair where we treat the user review as a query and the target item reviews as a document, and we denote the function as $m(\cdot, \cdot)$. Formally, each user review r_k^u can be alternatively represented as a sequence of word embeddings $[\boldsymbol{e}_1^u, \dots, \boldsymbol{e}_M^u] := \boldsymbol{S}_u^k$, and the item document is a concatenation of a sequence of word embeddings of its each review $[\boldsymbol{e}_1^i, \dots, \boldsymbol{e}_M^i, \dots, \boldsymbol{e}_{(N-1)M+1}^i, \dots, \boldsymbol{e}_{NM}^i] := \boldsymbol{S}_i$, where $\boldsymbol{e}_k^u,\ \boldsymbol{e}_k^i \in \mathbb{R}^{d_w}$, $\boldsymbol{S}_u^k \in \mathbb{R}^{d_w \times M}$, $\boldsymbol{S}^i \in \mathbb{R}^{d_w \times MN}$, d_w is the dimension of the word embedding, M is the review length, and N is the number of review from the target item. To get the relevant matching score from the k-th user review to its target item, we first compute the word similarity matrix S:

$$\boldsymbol{M} = \boldsymbol{S}_u^{k^T} \boldsymbol{S}_i \in \mathbb{R}^{M \times MN} \tag{7}$$

where $\mathbf{M}_{i,j}$ can be considered as similarity score by matching i-th word of user review with j-th word of item document. We apply mean pooling and max pooling on every row of similarity matrix to obtain discriminate features:

$$mean(\mathbf{M}) = \begin{bmatrix} mean(\mathbf{M_{1:}}) \\ \dots \\ mean(\mathbf{M_{n:}}) \end{bmatrix} \in \mathbb{R}^M \ , max(\mathbf{M}) = \begin{bmatrix} max(\mathbf{M_{1:}}) \\ \dots \\ max(\mathbf{M}_{n:}) \end{bmatrix} \in \mathbb{R}^M \tag{8}$$

Also, we consider the relative important score for each word in the user review $\boldsymbol{S}_u^k$ by applying a function $imp(\cdot)$:

$$imp(\mathbf{S}_u^k) = \begin{bmatrix} imp(\mathbf{e}_l^u) \\ \dots \\ imp(\mathbf{e}_l^u) \end{bmatrix} \in \mathbb{R}^M \quad \text{where,} \quad imp(\boldsymbol{e}_j^u) = \frac{\exp(\boldsymbol{w}_p^T \boldsymbol{e}_k^u)}{\sum_{o=1}^n \exp(\boldsymbol{w}_p^T \boldsymbol{e}_o^u)} \tag{9}$$

where $\boldsymbol{w}_p \in \mathbb{R}^{d_w}$, then the input feature for scoring function parameterized by a 2 layer feed-forward neural network is:

$$\boldsymbol{I}^{rel} = \begin{bmatrix} imp(\boldsymbol{S}_u^k) \odot mean(\mathbf{M}) \\ imp(\boldsymbol{S}_u^k) \odot max(\mathbf{M}) \end{bmatrix} \in \mathbb{R}^{2M} \tag{10}$$

Hence the relevant score between the k-th user review $\boldsymbol{S}_k^u$ and item document $\boldsymbol{S}^i$ is:

$$m(\boldsymbol{S}_k^u, \boldsymbol{S}^i) = \mathrm{FFN}\big(\mathrm{FFN}(\boldsymbol{I}^{rel})\big) = \alpha_k \in [-\infty, \infty] \tag{11}$$

When there is no user review relevant to the target item, we can expect each relevant score $\alpha_k \ll 0$. However, if we naively normalized the relevant scores, and use them as weights to measure the importance of each user review, the final dynamic user representation we get by weighted sum of the user review representations would be a non-zero vector. It is due to the fact that after the normalized process, every relevant score will be assigned as a probability measure, and summation of these probabilities being 1 makes the situation that every normalized relevant weight $a_k \approx 0$ become impossible, hence the weighted sum of the user reviews cannot be a zero vector. For example, suppose that the relevant scores of all user reviews are $\alpha_k = -100$, where $k = 1, \dots N$, then the normalized relevant score would be $\alpha_k = \frac{1}{N}$. The dynamic item representation will become $\boldsymbol{u}^{dynamic} = \sum_{k=1}^{N} \frac{1}{N} \boldsymbol{r}_k^u$, which is not a zero vector even when all user reviews are not relevant to the target item. To resolve the problem, we use the zero-attention network motivated by [1].

Zero-Attention Network: we introduce a zero score $\alpha_0 = 0$, and re-normalize the relevant scores by taking the zero score in to account. Formally, $\hat{\alpha}_k = \frac{\exp(0)+\exp(\alpha_k)}{\exp(0)+\exp(\alpha_1)+\dots\exp(\alpha_m)} = \frac{1+\exp(\alpha_k)}{1+\exp(\alpha_1)+\dots\exp(\alpha_m)}$, $k = 1, \dots N$, and $\hat{\alpha}_0 = \frac{1}{1+\exp(\alpha_1)+\dots\exp(\alpha_m)}$, then the user dynamic representation is,

$$\mathbf{u}^{dynamic} = \sum_{k=1}^{N} \hat{\alpha}_k \mathbf{r}_k^u + \hat{\alpha_0} \vec{\mathbf{0}} \tag{12}$$

Intuitively, when $\alpha_k \ll 0$, the normalized score $\hat{\alpha}_k \approx 0$ for all $k = 1, \dots, N$, and the $\boldsymbol{u}^{dynamic} \approx \vec{\mathbf{0}}$, which is close to a zero vector. In the other hand, if there exist a large relevant score, for example $\alpha_k = 10$ for a certain k, the effect of $\alpha_0 = 0$ will be very low, and the normalized score will be $\hat{\alpha}_k \approx 1$, and $\boldsymbol{u}^{dynamic} \approx \boldsymbol{r}_k^u$

3.4 Prediction Layer

This layer combine the static and dynamic user representations to form a final user representation learned from reviews. Also, it learns a final item static representation from reviews by 1-layer feed-forward neural network:

$$\boldsymbol{u}^r = \text{Relu}\Big(\Big[\mathbf{W}_u^{static}, \mathbf{W}_u^{dynamic}\Big] \begin{bmatrix} \boldsymbol{u}^{static} \\ \boldsymbol{u}^{dynamic} \end{bmatrix} + \mathbf{b}_u \Big) \tag{13}$$

$$\boldsymbol{i}^r = \text{Relu}\Big(\boldsymbol{W}_i^{static} \boldsymbol{i}^{static} + \mathbf{b}_i \Big) \tag{14}$$

where $\boldsymbol{W}_u^{static},\ \boldsymbol{W}_u^{static},\ \boldsymbol{W}_i^{dynamic},\ \boldsymbol{W}_i^{static} \in \mathbb{R}^{d_h \times d_r}$, $\boldsymbol{b}_u,\ \boldsymbol{b}_i \in \mathbb{R}^{d_h}$. Finally, we combine the user and item id embeddings $\boldsymbol{u}^{id},\ \boldsymbol{i}^{id} \in \mathbb{R}^{d_h}$, with user and item embeddings learned from reviews $\boldsymbol{u}^r,\ \boldsymbol{i}^r$, to form their final representations, which are $\boldsymbol{u} = \boldsymbol{u}^r + \boldsymbol{u}^{id},\ \boldsymbol{i} = \boldsymbol{i}^r + \boldsymbol{i}^{id}$. We take the user and item embeddings as input to get the final rating prediction:

$$\hat{y}_{u,i} = \boldsymbol{w}_f^T (\boldsymbol{u} \odot \boldsymbol{i}) + b_u + b_i + b_g \tag{15}$$

where $\boldsymbol{w}_f \in \mathbb{R}^{d_h},\ b_u,\ b_i,\ b_g \in \mathbb{R}$

3.5 Training Objective

Besides a regression loss for the rating prediction, an auxiliary loss is utilized for better training the relevance matching function. Specifically, we assumed there is a user-item pair (u, i) with the *ground-truth* rating $y_{u,i}$ in the training stage, and the *ground-truth* review $r^g_{u,i}$ written from the user to the target item is treated as a "positive query" to the target item. Also we randomly sample a review from the different user different item as a "negative query" to the target item which is $r^n_{u,i}$. The corresponding word sequence representation of the *ground-truth* review, negative review and target item document is $\boldsymbol{S}^u_g,\ \boldsymbol{S}^u_n,\ \boldsymbol{S}^i$. Ideally, a good relevance matching function $m(\cdot, \cdot)$ can distinguish the positive query-document pair from the negative one, in other words, we wish $m(\boldsymbol{S}^u_g, \boldsymbol{S}^i) > m(\boldsymbol{S}^u_n, \boldsymbol{S}^i)$. In the same time, we want to minimize the regression loss between ground-truth rating $y_{u,i}$ and predicted rating $\hat{y}_{u,i}$ computed from Equation 15. To achieve the above two goals, we write the objective function as followed,

$$loss = \sum_{\{(u,i)\} \in \mathcal{S}} \underbrace{\Big(y_{u,i} - \hat{y}_{u,i} \Big)^2}_{\text{regression loss}} - \underbrace{\Big(\log\big(m(\boldsymbol{S}^u_g, \boldsymbol{S}^i)\big) + \log\big(1 - m(\boldsymbol{S}^u_n, \boldsymbol{S}^i)\big) \Big)}_{\text{auxiliary loss}}$$

4 Experimental Setup

Datasets and Evluation Metrics. We conduct our experiment on four different categories of 5-core Amazon product review datasets [13]. The statistics of these four categories are shown in the first and second columns of the Table 1. For each dataset, we randomly split user-item pairs into training, validation, and testing sets with ratio 8:1:1. We use NLTK[3] to tokenize sentences and words of reviews. We let the number of reviews be the same for profiling user and item where the number of reviews is set to cover 90% of users for the balance of efficiency and performance. We adopt Mean Square Error (MSE) as the main metric to evaluate the performance of our model. The source code can be found here [1].

[1] https://github.com/HansiZeng/ZARM

Compared Methods. To evaluate the performance of our method, we compare it to several state-of-the-art baseline models: (1) **MF** [17]: a basic but well-known CF model that predict the rating using inner product between user, item hidden representations plus user, item and global bias; (2) **NeurMF** [15]: the CF based model combines linearity of GMF and non-linearity of MLPs for modeling user and item latent representations; (3) **HFT** [19]: the topic modeling based model combines the ratings with reviews via LDA; (4) **DeepCoNN** [38]: the CNN based model uses two convolution neural network to learn user and item representation; (5) **NARRE** [5]: the CNN based model modifies the DeepCoNN by using the attention network over review-level to select reviews with more informativeness. (6) **MPCN** [29]: the model that selects informative reviews from user and item by review-level pointers using the co-attention technique, and selects informative word-level representations for the rating prediction by applying word-level pointers over the selected reviews; (7) **AHN** [10]: a dynamic model using co-attention mechansim but treats user and item asymmetrically; (8) **ZARM-static**: the variant of the ZARM that only user static representations; And (9) **ZARM-dynamic**: the variant of the ZARM that only uses user dynamic representations.

Parameter Settings. We use 300-dimensional pretrained word embeddings from Google News [20], and employ the Adam[16] for optimization with an inital learning rate 0.001. We set the dimension of sentence hidden vector and review hidden vector as 150, and the latent dimension of the prediction layer as 32. Also, the convolution kernel size is 1 or 3 based on the performance in each dataset, and number of head for each self-attention layer is 2. We apply dropout after the word embedding layer, after each feed forward layer in sequence encoding modules, and before the prediction layer with rate $[0.2, 0.5, 0.5]$. The hidden dimension of the two layer neural network in the Relevance Matching Module is set to 16. The hyper-parameters of baselines are set following the settings of their original papers.

5 Results and Analysis

The MSE results of compared models are shown in Table 1. Based on the results, we can make several observations. Firstly, the siamese models outperform the interaction-based models significantly. As discussed previously, due to the fact that not every user exists informative review to the target, interaction-based models that force to extract informative reviews from user data will suffer from heavily over-fitting. Among siamese networks, we observe that the ZARM-static outperforms the other siameses models. This demonstrates that ZARM-static can capture the review hierarchical structure and use attention neural network to select the important information in each level. Among interaction-based models, ZARM-dynamic outperforms the other baselines such as MPCN and AHN. This demonstrates the effectiveness of the relevance matching component in discovering relevant information from vast review text and the utility of the auxiliary training loss that makes the found relevant review more aligned with the *ground-truth* that reflect the user true opinion on the target item. Finally, our model

Table 1: Experiment results on benchmark datasets. † and ‡ represents the best performance among siamese and interaction-based models, respectively. The bold value is the best performance among all models in each dataset.

Dataset		Toys & Games	Video Games	Kindle Store	Office Products
#Reviews / #Users / #Items		167k / 19k / 12k	232k / 24k / 11k	983k / 68k / 62k	53k / 2k / 4k
Non-text-based	MF	0.8010	1.0979	0.6231	0.6954
	NeuMF	0.8012	1.0931	0.6255	0.6941
Siamese	HFT	0.7947†	1.0837	0.6172	0.6881
	DeepCoNN	0.8273	1.1241	0.6437	0.7102
	NARRE	0.7982	1.0881	0.6199	0.6794
	ZARM-static	0.7952	1.0774†	0.6159†	0.6757†
Interaction-based	MPCN	0.8199	1.1062	0.6337	0.7101
	AHN	0.8233	1.1137	0.6341	0.7341
	ZARM-dynamic	0.8054‡	1.1054‡	0.6279‡	0.7024‡
Hybrid	ZARM	**0.7881**	**1.0632**	**0.6083**	**0.6695**

Table 2: Ablation Study (validation MSE) on four datasets

Architecture	Toys-and-Games	Video-Games	Kindle-Store	Office-Product
Default	0.7897	**1.0611**	**0.5961**	**0.6731**
(1) max pooling	0.7922	1.0645	0.6075	0.6795
(2) avg embedding	**0.7854**	1.0641	0.6043	0.6742
(3) Remove pos. vec.	0.7913	1.0654	0.5985	0.6755
(4) Remove u/i bias	0.8021	1.0713	0.6022	0.6761
(5) Remove aux. loss	0.8147	1.0944	0.6189	0.6893
(6) Add item dyn.	0.7938	1.0695	0.6053	0.6800

(i.e., ZARM) shows consistently improvement over siamese and interaction-based models across all datasets. Our model uses the zero-attention network that can build dynamic user representations from reviews when there are high informative reviews and can easily degrade to user static representations when there is not. This strategy combines the advantages of both the siamese and interaction-based models.

5.1 Ablation Studies

We conduct the ablation study on the validation sets of the the four benchmark datasets. We report the performance of 6 variant models from the defualt model setting: (1) we change the static review aggregator in Equation 6 to max pooling; (2) we encode each review using average embedding of words; (3) We use the relative position representations [26] to encode the relative position between entities in the self-attention network, now we remove the position encoding vectors to conduct ablation study; (4) we remove the user and item bias in Equation 15; (5) we remove the auxiliary loss in the training objective; And (6) we make our user and item representations symmetrically by adding the dynamic item encoding to represent the item.

As shown in Table 2, the performance of ZARM would drop when we use max pooling in the aggregator, remove the position encoding vectors in self-attention network, or remove the u/i bias. Using average word embeddings for review embeddings achieves suboptimal performance on most datasets, but it

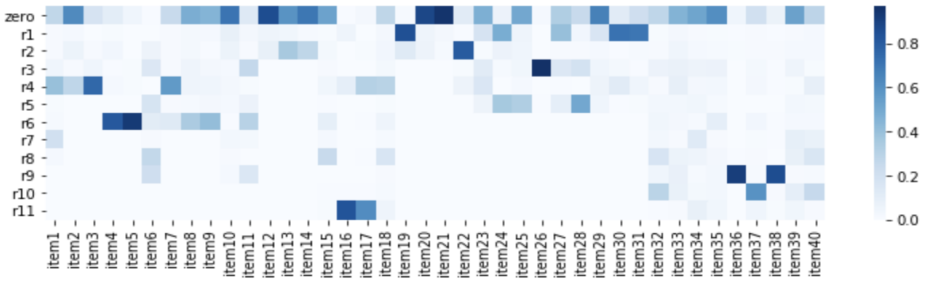

Fig. 3: The distribution of relevance matching scores of each user review given a user-item pair.

also outperforms the default ZARM on Toys-and-Games, which indicates that such simple aggregators may have some value on specific data types. Interestingly, in our experiments, the variant architecture that encodes item using its dynamic and static representation underperforms the default ZARM which only use the item static encoding. This indicates that building interaction-based representations on the item side may not as profitable as they are on the user side, or the current interaction module is not suitable for the construction of dynamic item representations.

5.2 Behavior of the Dynamic Interaction Matching

We conduct several experiments on investigating the behaviors of the dynamic interaction matching Firstly, we investigate the number of relevant reviews ($\alpha_k > 0$ in Eq. (11)) each user have to the target item as shown in Figure 2. Although the number of relevant review from the user to the target is dataset dependent, there are around 50% of user-item pairs do not have the relevant review in each dataset where the type of pairs in Video Games dataset account for 60% the most, and in Office Product account for 48% the least. On the other hand, some users have more than one relevant reviews to their target items. For example, in the Toys & Games dataset, 15% of the user-item pairs have relevant review more than 3. Such observation implies that some users have consistent interests and tend to buy items with similar characteristics, which lead to their target item matched to her multiple history items in high possibility.

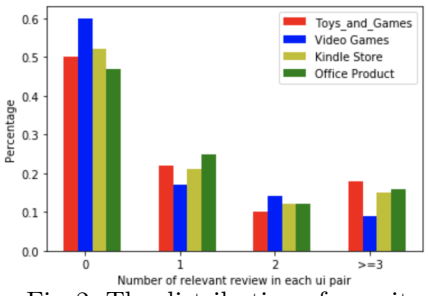

Fig. 2: The distribution of user-item pairs with different numbers of relevant review ($\alpha_k > 0$).

To further analyze the interaction module in our model, we randomly sample 40 users and their corresponding target items in the validation set for case studies. For each user we visualize the zero score and the relevant score of each review to the target item (from r1 to r11) as shown in Figure 3. We observe that there are roughly half of the user-item pairs having large zero score which is

Table 3: Examples of high relevant reviews r6, r9 and their corresponding target item documents. The first column is their complete user review, and the second column are sampled text selected from the item documents.

User Review	Target Item Document
I have bought other remote **control helicopters** only to take them outside and have a little breeze of wind **knock it down and break**. With the **Gyro Hercules** it can **truly withstand a hard fall** so you can fly it nearly anywhere.	My kids demolish other helicopters / **keeps on going and not falls down** / helicopter / **Gyro Hercules helicopter** / it is durable enough I can't even break it with my terrible skills.
Bought for our **Granddaughter(she is 3)** for **Christmas**. She just loves the write on wipe off A,B,C's and 1,2,3's. The **art projects** that were included and quality of the items for the project, TERRIFIC! Would recommend for all 3 year olds.	This is perfect for a rainy day **Christmas Vacation** / **my three yr old LOVES crafts** / Filled with all the supplies to make 16 high quality crafts

larger than 0.5. On the other hand, there are some users containing reviews with high relevant scores like the pair (user5, item5), (user36, item36) with review id r6, r9. We then take a closer look into the two high relevant review r6, r9 and their corresponding target item documents as shown in Table 3. We observe that the high relevant reviews and their target item documents share multiple similar keywords, and these keywords are highly informative that can describe the item characteristics to a large extent. For example, the keyword "Gyro Hercules" in the r6 and "Gyro Hercules helicopter" in its corresponding target item document have high textual similarity and describe the general characteristics of the two items that that r6 and target item belong to. Moreover, The user true opinion on the target item can be reflected in the high relevant reviews from the user to target. For example, the first target item (item5) which has the advantage of "keep on going and not falls down" meets the user interest that is shown in r6 that mentions she likes a helicopter that is "truly withstand a hard fall". And the second target item (item 36) which is suitable for kid Christmas gift conforms to the user interest reflected in r9 in which she mention that she needs a Christmas gift for her 3-year-old granddaughter.

6 Conclusion

We propose a new model ZARM for the review based rating prediction task. In our model, the interaction module based on relevance matching function with zero-attention network is utilized to learn user dynamic representation in more flexible way. And the auxiliary loss plugged into the training object make the relevance matching function better trained. Experiments on the four Amazon benchmark datasets show our model can outperform the state-of-art models based on the siamese network and interaction-based network. By conducting case studies, we take a deeper look into the behavior of our interaction module, and investigate the several statistical and semantic characteristics of the relevant reviews for users to targets extracted by the interaction module.

Acknowledgements

This work was supported in part by the School of Computing, University of Utah and in part by NSF IIS-2007398. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

1. Ai, Q., Hill, D.N., Vishwanathan, S., Croft, W.: A zero attention model for personalized product search. Proceedings of the 28th ACM International Conference on Information and Knowledge Management (2019)
2. Bi, K., Ai, Q., Croft, W.B.: A transformer-based embedding model for personalized product search. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3397271.3401192, `https://doi.org/10.1145/3397271.3401192`
3. Bird, S.: Nltk: The natural language toolkit. ArXiv **cs.CL/0205028** (2006)
4. Catherine, R., Cohen, W.: Transnets: Learning to transform for recommendation. Proceedings of the Eleventh ACM Conference on Recommender Systems (2017)
5. Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: Proceedings of the 2018 World Wide Web Conference (2018)
6. Chen, Q., Zhu, X.D., Ling, Z., Wei, S., Jiang, H., Inkpen, D.: Enhanced lstm for natural language inference. In: ACL (2017)
7. Cong, D., Zhao, Y., Qin, B., Han, Y., Zhang, M., Liu, A., Chen, N.: Hierarchical attention based neural network for explainable recommendation. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval (2019)
8. Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? a worrying analysis of recent neural recommendation approaches. Proceedings of the 13th ACM Conference on Recommender Systems (2019)
9. Diao, Q., Qiu, M., Wu, C.Y., Smola, A., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In: KDD '14 (2014)
10. Dong, X., Ni, J., Cheng, W., Chen, Z., Zong, B., Song, D., Liu, Y., Chen, H., Melo, G.: Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation. In: AAAI (2020)
11. Guo, J., Fan, Y., Ai, Q., Croft, W.: A deep relevance matching model for ad-hoc retrieval. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (2016)
12. Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., Wu, C., Croft, W.B., Cheng, X.: A deep look into neural ranking models for information retrieval. Information Processing & Management **57**(6), 102067 (2020)
13. He, R., McAuley, J.: Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. ArXiv **abs/1602.01585** (2016)
14. He, X., Chen, T., Kan, M., Chen, X.: Trirank: Review-aware explainable recommendation by modeling aspects. In: CIKM '15 (2015)
15. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. Proceedings of the 26th International Conference on World Wide Web (2017)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2015)
17. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42** (2009)
18. Ling, G., Lyu, M.R., King, I.: Ratings meet reviews, a combined approach to recommend. In: RecSys '14 (2014)
19. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: RecSys '13 (2013)

20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. ArXiv **abs/1310.4546** (2013)
21. Rao, J., Liu, L., Tay, Y., Yang, W., Shi, P., Lin, J.: Bridging the gap between relevance matching and semantic matching for short text similarity modeling. In: EMNLP/IJCNLP (2019)
22. Ren, Z., Liang, S., Li, P., Wang, S., Rijke, M.: Social collaborative viewpoint regression with explainable recommendations. In: WSDM '17 (2017)
23. Sachdeva, N., McAuley, J.: How useful are reviews for recommendation? a critical review and potential improvements. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020)
24. Santos, C.D., Tan, M., Xiang, B., Zhou, B.: Attentive pooling networks. ArXiv **abs/1602.03609** (2016)
25. Seo, S., Huang, J., Yang, H., Liu, Y.: Interpretable convolutional neural networks with dual local and global attention for review rating prediction. Proceedings of the Eleventh ACM Conference on Recommender Systems (2017)
26. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. ArXiv **abs/1803.02155** (2018)
27. Tan, Y., Zhang, M., Liu, Y., Ma, S.: Rating-boosted latent topics: Understanding users and items with ratings and reviews. In: IJCAI (2016)
28. Tay, Y., Luu, A.T., Hui, S.: Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In: EMNLP (2018)
29. Tay, Y., Luu, A.T., Hui, S.: Multi-pointer co-attention networks for recommendation. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018)
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. ArXiv **abs/1706.03762** (2017)
31. Wu, C., Wu, F., Liu, J., Huang, Y.: Hierarchical user and item representation with three-tier attention for recommendation. In: NAACL-HLT (2019)
32. Xiong, C., Dai, Z., Callan, J., Liu, Z., Power, R.: End-to-end neural ad-hoc ranking with kernel pooling. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (2017)
33. Yang, R., Zhang, J., Gao, X., Ji, F., Chen, H.: Simple and effective text matching with richer alignment features. ArXiv **abs/1908.00300** (2019)
34. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: HLT-NAACL (2016)
35. Zeng, H., Ai, Q.: A hierarchical self-attentive convolution network for review modeling in recommendation systems. arXiv preprint arXiv:2011.13436 (2020)
36. Zhang, W., Yuan, Q., Han, J., Wang, J.: Collaborative multi-level embedding learning from reviews for rating prediction. In: IJCAI (2016)
37. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval (2014)
38. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (2017)