Asymmetrical Hierarchical Networks with Attentive Interactions for Interpretable Review-Based Recommendation

Xin Dong,[1]* Jingchao Ni,[2]‡ Wei Cheng,[2] Zhengzhang Chen,[2] Bo Zong,[2] Dongjin Song,[2] Yanchi Liu,[2] Haifeng Chen,[2] Gerard de Melo[1]
[1]Rutgers University, [2]NEC Laboratories America
[1]xd48@rutgers.edu, gdm@demelo.org
[2]{jni, weicheng, zchen, bzong, dsong, yanchi, haifeng}@nec-labs.com

Abstract

Recently, recommender systems have been able to emit substantially improved recommendations by leveraging user-provided reviews. Existing methods typically merge all reviews of a given user (item) into a long document, and then process user and item documents in the same manner. In practice, however, these two sets of reviews are notably different: users' reviews reflect a variety of items that they have bought and are hence very heterogeneous in their topics, while an item's reviews pertain only to that single item and are thus topically homogeneous. In this work, we develop a novel neural network model that properly accounts for this important difference by means of asymmetric attentive modules. The user module learns to attend to only those signals that are relevant with respect to the target item, whereas the item module learns to extract the most salient contents with regard to properties of the item. Our multi-hierarchical paradigm accounts for the fact that neither are all reviews equally useful, nor are all sentences within each review equally pertinent. Extensive experimental results on a variety of real datasets demonstrate the effectiveness of our method.

1 Introduction

The rapid shift from traditional retail and services to online transactions has brought forth a large volume of review data in areas such as e-commerce, dining, tourism, among many others. While such reviews are routinely consulted directly by consumers and affect their decision making, recent work has shown that they can also be exploited by intelligent algorithms. The detailed semantic cues that they harbor not only reveal different aspects (*e.g.*, quality, material, color, and *etc.*) of an item, but also reflect the sentiment of users towards these aspects. Such fine-grained signals are extremely valuable to a recommender system and significantly complement the sparse rating and click-through data, based on which many traditional collaborative filtering methods (Koren, Bell, and Volinsky 2009) have been developed. Thus, there has been a series of studies seeking to harness the potential of reviews in improving the recommendation quality

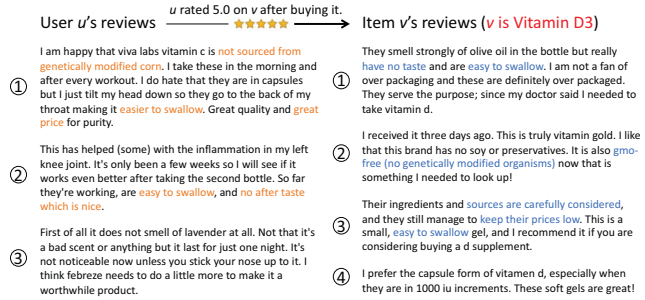

Figure 1: An example of user's and item's historical reivews. User u rated a 5.0 score on item v after purchasing it.

(Zheng, Noroozi, and Yu 2017; Catherine and Cohen 2017; Seo et al. 2017; Chen et al. 2018).

These studies have shown that leveraging reviews can indeed boost the recommendation effectiveness quite remarkably. Typically, these methods associate users with the respective sets of reviews they have written, while associating each item with the set of all reviews that have been written for it. To predict the rating for an unseen user–item pair, in a first step, the embeddings of that user and item are inferred from the respective sets of reviews via a neural network. Then, the two embeddings are matched to predict a numeric rating between them. For example, DeepCoNN (Zheng, Noroozi, and Yu 2017) relies on convolutional neural networks to learn user (item) embeddings, and on a factorization machine (Rendle 2010) to predict ratings. D-ATT (Seo et al. 2017) uses dual-attention based networks to learn embeddings, and a simple dot product to predict ratings.

Despite the encouraging progress, existing methods all regard the set of reviews by a user and the set of reviews for an item as the same type of documents, and invoke the same model (or even a shared model) to process them in parallel. In reality, however, the set of reviews for a user is fundamentally different from the set of reviews for an item. In particular, reviews for users correspond to a diverse set of items that they have rated, resulting in notably *heterogeneous* textual contents with a variety of topics for different items. In contrast, each item's reviews are only about itself, and the

*This work was done when the first author was an intern at NEC Laboratories America. ‡Corresponding author.

contents are thus *homogeneous* in the sense that the topic is limited to a single narrow domain. For example, Fig. 1 shows several reviews from Amazon's health domain. User u's historical reviews describe three items, Vitamin C, anti-inflammatory medication, and an air freshener, while all reviews for item v are about itself, *i.e.*, Vitamin D3.

This profound difference necessitates distinct forms of attention to be paid on user reviews as opposed to item reviews, when deciding whether to recommend an item v to a user u. To predict u's preference of v, it is important to extract from u's reviews those aspects that pertain most to v, e.g., comments on items that are similar to v. In contrast, from v's reviews, we wish to account for the sentiment of other users with regard to relevant aspects of v. If u pays special attention to certain aspects of items similar to v, while other users wrote highly about v with regard to these particular aspects, then it is much more likely that v will be of interest to u. For example, in Fig. 1, reviews 1 and 2 of u are about non-prescription medicines that are similar to v. In reviews 1 and 2, u mentioned aspects such as "not sourced from genetically modified corn", "easier to swallow", "great price", and "no after taste", indicating that u considers the source and price and prefers easily swallowed products without after-taste. Meanwhile, reviews 1-3 of v mention that v "have no taste", is "easy to swallow", "gmo-free", and "prices low", which are opinions expressed by others that match u's preferences. Thus, v is likely to be of interest to u, and u indeed marked a 5.0 score on v after purchasing it.

Another vital challenge in our problem is how to reliably represent each review. It is worth noting that sentences are not equally useful within each review. For example, in Fig. 1, the 2nd sentence in u's review 1, "I take these in the morning and after every workout." conveys little regarding u's concerns for Vitamin C, and thus is less pertinent than other sentences in the same review. Since including irrelevant sentences can introduce noise and may harm the final embedding quality, it is crucial to aggregate only useful sentences to represent each review.

To address the above challenges, in this paper, we propose an Asymmetrical Hierarchical Network with Attentive Interactions (AHN) for recommendation. AHN progressively aggregates salient sentences to induce review representations, and aggregates pertinent reviews to induce user (item) representations. AHN is particularly characterized by its asymmetric attentive modules to flexibly distinguish the learning of user embeddings as opposed to item embeddings. For items, several attention layers are invoked to highlight sentences and reviews that contain rich aspect and sentiment information. For users, we designed an interaction-based co-attentive mechanism to dynamically select a homogeneous subset of contents related to the current target item. In this manner, AHN hierarchically induces embeddings for user–item pairs reflecting the most useful knowledge for personalized recommendation. In summary, our contributions are

- We identify the asymmetric attention problem for review-based recommendation, which is important but neglected by existing approaches.
- We propose AHN, a novel deep learning architecture that not only captures both of the asymmetric and hierarchical characteristics of the review data, while also enabling interpretability of the results.
- We conduct experiments on 10 real datasets. The results demonstrate that AHN consistently outperforms the state-of-the-art methods by a large margin, meanwhile providing good interpretations of the predictions.

2 Related Work

Exploiting reviews in learning user and item representations has been proven considerably useful in recent work on recommendation. Many methods primarily focus on topic modeling based on the review texts. For example, HFT (McAuley and Leskovec 2013) employs LDA to discover the latent aspects of users and items from reviews. RMR (Ling, Lyu, and King 2014) extracts topics from reviews to enhance the user and item embeddings obtained by factorizing the rating matrix. TopicMF (Bao, Fang, and Zhang 2014) jointly factorizes a rating matrix and bag-of-words representations of reviews to infer user and item embeddings. Despite the improvements achieved, these methods only focus on topical cues in reviews, but neglect the rich semantic contents. Moreover, these methods typically represent reviews as bag-of-words, and thus remain oblivious of the order and contexts of words and sentences in reviews, which are essential for modeling the characteristics of users and items (Zheng, Noroozi, and Yu 2017).

Inspired by the astonishing advancements of recent deep NLP techniques in various applications (Santos et al. 2016; Wang et al. 2018; Peters et al. 2018; Dong and De Melo 2018; Devlin et al. 2018; Yang et al. 2019), there has been increasing interests in studying deep learning models. For example, DeepCoNN (Zheng, Noroozi, and Yu 2017) employs CNNs as an automatic feature extractor to encode each user (item) into a low-dimensional vector by parsing the relevant set of historical reviews. TransNet (Catherine and Cohen 2017) extends DeepCoNN by augmenting the CNN architecture with a multi-task learning scheme to regularize the user and item embeddings towards the target review. These methods, however, lack interpretability in their results.

To better understand the predictions, several attention-based methods have been developed. D-ATT (Seo et al. 2017) incorporates two kinds of attention mechanisms on the words of reviews to find informative words. NARRE (Chen et al. 2018) invokes review-level attention weights to aggregate review embeddings to form user (item) embeddings. HUITA (Wu et al. 2019) is equipped with a symmetric hierarchical structure, where, at each level (e.g., word level), a regular attention mechanism is employed to infer the representation of the subsequent level (e.g., sentence level). MPCN (Tay, Luu, and Hui 2018) models the interactions between a user's reviews and an item's reviews via co-attention based pointers that are learned with the Gumbel-Softmax trick (Jang, Gu, and Poole 2016). However, all these methods just learn user and item embeddings in parallel and fail to consider the important differences between the two. As discussed before, this leads to suboptimal predictions.

Unlike the aforementioned methods, our method learns

several hierarchical aggregators to infer user (item) embeddings. The aggregators are asymmetric to flexibly pay varying levels of attention to user's (item's) reviews, so as to enhance the prediction accuracy and model interpretability.

3 Our Proposed Model

In this section, we introduce our AHN model in a bottom-up manner. Fig. 2 illustrates the architecture of AHN.

Sentence Encoding

The sentence encoding layer (omitted in Fig. 2) aims to transform each sentence (in each review) from a sequence of discrete word tokens to a continuous vector embedding. We use a word embedding model to lay the foundation of this layer. Suppose the sentence s has l words. By employing a word embedding matrix $\mathbf{E} \in \mathbb{R}^{d \times |\mathcal{V}|}$, s can be represented by a sequence $[\mathbf{e}_1, ..., \mathbf{e}_l]$, where $\mathbf{e}_i$ is the embedding of the i-th word in s, d is the dimensionality of the word embedding, and $\mathcal{V}$ is the whole vocabulary of words. The matrix $\mathbf{E}$ can be initialized using word embeddings such as word2vec (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014), which are widely used in NLP. To refine the word embeddings, $\mathbf{E}$ is fine-tuned during model training.

To learn an embedding for s, we employ a bi-directional LSTM (Peters et al. 2018) on its constituent word embeddings, and apply max-pooling on the hidden states to preserve the most informative information. That is

$$\mathbf{s} = \max([\tilde{\mathbf{e}}_1, ..., \tilde{\mathbf{e}}_l]), \tag{1}$$

where $\mathbf{s}$ is the embedding of s and

$$\tilde{\mathbf{e}}_i = \text{BiLSTM}(\tilde{\mathbf{e}}_{i-1}, \mathbf{e}_i) \ \ (1 \le i \le l), \tag{2}$$

where $\tilde{\mathbf{e}}_0$ is initialized by an all-zero vector $\mathbf{0}$.

Suppose a review has k sentences. We can then represent this review by a sequence $[\mathbf{s}_1, ..., \mathbf{s}_k]$, where $\mathbf{s}_i$ is the embedding of the i-th sentence in the review, as inferred by Eq. (1). However, using Eq. (1), each $\mathbf{s}_i$ only encodes its own semantic meaning, but remains oblivious of any contextual cues from its surrounding sentences in the same review. To further refine the sentence embedding, we introduce a context-encoding layer by employing another bi-directional LSTM on top of the previous layer to model the temporal interactions between sentences, i.e.,

$$\tilde{\mathbf{s}}_i = \text{BiLSTM}(\tilde{\mathbf{s}}_{i-1}, \mathbf{s}_i) \ \ (1 \le i \le k), \tag{3}$$

where $\tilde{\mathbf{s}}_i$ is the final embedding of the i-th sentence in the review and $\tilde{\mathbf{s}}_0$ is initialized as $\mathbf{0}$.

Sentence-Level Aggregation

Next, we develop sentence-level aggregators to embed each review into a compact vector from its constituent sentences. As discussed before, an ideal method should learn review embeddings in an asymmetric style. Thus, we design AHN to learn different attentive aggregators for users and items, respectively, as highlighted in Fig. 2.

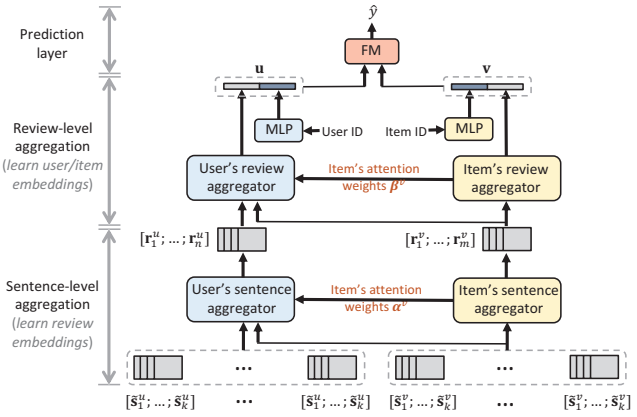

Figure 2: The overall architecture of AHN.

Sentence Aggregator for Items. Given an item, we are interested in sentences that contain other users' sentiments on different aspects of the item, which are the key factors to determine its overall rating. To build an informative embedding for each review upon such sentences, we use a sentence-level attention network to aggregate the sentence embeddings $[\tilde{\mathbf{s}}_1^v, ..., \tilde{\mathbf{s}}_k^v]$ as follows, where the superscript v is used to distinguish an item's notation from a user's notation.

$$\mathbf{r}^v = \sum_{i=1}^{k} \alpha_i^v \tilde{\mathbf{s}}_i^v, \tag{4}$$

Here, $\sum_{i=1}^{k} \alpha_i^v = 1$, and α_i^v is the attention weight assigned to sentence $\tilde{\mathbf{s}}_i^v$. It quantifies the informativeness of sentence $\tilde{\mathbf{s}}_i^v$ with respect to v's overall rating, compared to other sentences. The weights α_i^v's are computed by our attentive module, which takes the sentence embeddings as the input and is given by

$$\alpha_i^v = \frac{\exp\left(\mathbf{v}_s^\top (\tanh\left(\mathbf{W}_s \tilde{\mathbf{s}}_i^v\right) \otimes \sigma(\hat{\mathbf{W}}_s \tilde{\mathbf{s}}_i^v))\right)}{\sum_{j=1}^{k} \exp\left(\mathbf{v}_s^\top (\tanh\left(\mathbf{W}_s \tilde{\mathbf{s}}_j^v\right) \otimes \sigma(\hat{\mathbf{W}}_s \tilde{\mathbf{s}}_i^v))\right)}. \tag{5}$$

Here, $\mathbf{v}_s \in \mathbb{R}^{h \times 1}$, $\mathbf{W}_s \in \mathbb{R}^{h \times d}$, and $\hat{\mathbf{W}}_s \in \mathbb{R}^{h \times d}$ are parameters, $\otimes$ is the element-wise product, and $\sigma(\cdot)$ is the sigmoid function. As suggested by (Ilse, Tomczak, and Welling 2018), the approximate linearity of $\tanh(\cdot)$ in $[-1, 1]$ could limit the expressiveness of the model, and this problem can be alleviated by introducing a non-linear gating mechanism. Thus, in Eq. (5), a gate function $\sigma(\hat{\mathbf{W}}_s \tilde{\mathbf{s}}_i^v)$ is incorporated, which is indeed found effective in our experiments.

Sentence Aggregator for Users. Next, we develop an interaction-based sentence aggregator for users. Given a user–item pair, we aim to select a homogeneous subset of sentences from each of the user's reviews such that the selected sentences are relevant to the item to be recommended, i.e., the *target item*. In the following, we introduce a co-attentive network that uses the target item's sentences to guide the search of user's sentences.

After the sentence encoding layer, we can represent each review by a matrix $\mathbf{R} = [\tilde{\mathbf{s}}_1; ...; \tilde{\mathbf{s}}_k] \in \mathbb{R}^{d \times k}$, where $[\cdot;\cdot]$ is

the concatenation operation. Suppose a user has n reviews and an item has m reviews. Our method first concatenates all sentences of the item to form $[\mathbf{R}_1^v; ...; \mathbf{R}_m^v] \in \mathbb{R}^{d \times mk}$, whose constituent sentences are all relevant to the target item, and thus can be used to guide the search of similar sentences from the user's reviews. To this end, we iterate over each $\mathbf{R}_i^u$ $(1 \leq i \leq n)$ to calculate an affinity matrix as follows, where the superscript u indicates user's notation.

$$\mathbf{G}_i = \phi(f(\mathbf{R}_i^u)^\top \mathbf{M}_s f([\mathbf{R}_1^v; ...; \mathbf{R}_m^v])), \quad (1 \leq i \leq n) \quad (6)$$

Here, $\mathbf{M}_s \in \mathbb{R}^{d_s \times d_s}$ is a learnable parameter, $\phi(\cdot)$ is an activation function such as ReLU, and $f(\cdot)$ is a mapping function such as a multi-layer perceptron (MLP). If $f(\cdot)$ is an identity mapping, Eq. (6) becomes a bilinear mapping. Here, the (p, q)-th entry of $\mathbf{G}_i$ represents the affinity between the p-th sentence of $\mathbf{R}_i^u$ and the q-th sentence of $[\mathbf{R}_1^v; ...; \mathbf{R}_m^v]$.

To measure how relevant the p-th sentence of the user's review $\mathbf{R}_i^u$ is to the target item, we use the maximum value in the p-th row of $\mathbf{G}_i$. The intuition is that, if a user's sentence (i.e., a row of $\mathbf{G}_i$) has a large affinity to at least one sentence of the target item (i.e., a column of $\mathbf{G}_i$) – in other words, the maximal affinity of this row is large – then this user's sentence is relevant to the target item.

However, not all sentences of the target item are useful for searching relevant sentences from the user. For instance, in Fig. 1, the first sentence of the item's review 2, "I received it three days ago." conveys little information about the target item, and hence cannot aid in identifying relevant sentences from the user, and indeed may introduce noise into the affinity matrix. To solve this problem, recall that α_i^v in Eq. (5) represents how informative an item's sentence is. Thus, we concatenate α_i^v's of all sentences of the target item to form $\boldsymbol{\alpha}^v \in \mathbb{R}^{1 \times mk}$. Subsequently, we compute an element-wise product between each row of $\mathbf{G}_i$ and the vector $\boldsymbol{\alpha}^v$, i.e., $\mathbf{G}_i \otimes_{\text{row}} \boldsymbol{\alpha}^v$. In this manner, the (p, q)-th entry, $(\mathbf{G}_i \otimes_{\text{row}} \boldsymbol{\alpha}^v)_{pq}$, is high only if the p-th sentence of the user is similar to the q-th sentence of the target item and the q-th sentence of the target item is non-trivial.

By summarizing the above insights, we learn attention weights for the sentences in $\mathbf{R}_i^u$ for each $i \in [1, n]$ by

$$\boldsymbol{\alpha}_i^u = \text{softmax}(\max_{\text{row}}(\mathbf{G}_i \otimes_{\text{row}} \boldsymbol{\alpha}^v)), \quad (7)$$

where $\max_{\text{row}}$ refers to row-wise max-pooling for obtaining the maximum affinity. Intuitively, $(\boldsymbol{\alpha}_i^u)_j$ is large if the j-th sentence in the i-th review of the user describes some aspects of some item that is highly similar to the target item. This serves our purpose for selecting a homogeneous subset of sentences from the user.

Next, we use $\boldsymbol{\alpha}_i^u$ to aggregate the sentences in $\mathbf{R}_i^u$ to infer an embedding of the i-th review for the user:

$$\mathbf{r}_i^u = \sum_{j=1}^{k} (\boldsymbol{\alpha}_i^u)_j (\mathbf{R}_i^u)_{*j}, \quad (8)$$

where $(\mathbf{R}_i^u)_{*j}$ is the j-th column of $\mathbf{R}_i^u$. Recall that $\mathbf{R}_i^u = [\tilde{\mathbf{s}}_1^u; ...; \tilde{\mathbf{s}}_k^u]$, where each column of $\mathbf{R}_i^u$ is a sentence embedding. Note that our method iterates over i for $i \in [1, n]$ to calculate all review embeddings $\mathbf{r}_1^u$, ..., $\mathbf{r}_n^u$.

Remark. Our co-attentive mechanism employs the idea of sequence pair modeling but notably differs from the conventional co-attention used in QA systems (Santos et al. 2016; Xiong, Zhong, and Socher 2017; Zhang et al. 2017). First, we only consider one side of the affinity matrix, i.e., the user. Second, our affinity matrix is adapted by row-wise multiplication of $\boldsymbol{\alpha}^v$ to quantify the utility of the item's sentences. Thus, our method is designed specifically for learning asymmetric attentions from user–item interactions.

Review-Level Aggregation

From Eq. (4), we obtain review embeddings for an item, $\mathbf{r}_1^v$, ..., $\mathbf{r}_m^v$. From Eq. (8), we obtain review embeddings for a user, $\mathbf{r}_1^u$, ..., $\mathbf{r}_n^u$. As shown in Fig. 2, based on these review embeddings, we develop review-level aggregators to infer an embedding for each user and item, respectively.

As discussed before, different reviews exhibit different degrees of informativeness in modeling users and items. In particular, an item's reviews are homogeneous. Thus, we are interested in reviews with rich descriptions regarding its relevant aspects and corresponding sentiments, such as the reviews 1–3 of v in Fig. 1, compared with the less informative review 4 of v. To attend to such reviews, similar to Eq. (4), we aggregate the review embeddings to represent an item by

$$\tilde{\mathbf{v}} = \sum_{i=1}^{m} \beta_i^v \mathbf{r}_i^v, \quad (9)$$

where $\sum_{i=1}^{m} \beta_i^v = 1$, and β_i^v is the attention weight assigned to review $\mathbf{r}_i^v$. It quantifies the informativeness of the review $\mathbf{r}_i^v$ with respect to v's overall rating. β_i^v is produced by an attentive module with gating mechanism as follows:

$$\beta_i^v = \frac{\exp\left(\mathbf{v}_r^\top (\tanh\left(\mathbf{W}_r \mathbf{r}_i^v\right) \otimes \sigma(\hat{\mathbf{W}}_r \mathbf{r}_i^v))\right)}{\sum_{j=1}^{k} \exp\left(\mathbf{v}_r^\top (\tanh\left(\mathbf{W}_r \mathbf{r}_j^v\right) \otimes \sigma(\hat{\mathbf{W}}_r \mathbf{r}_i^v))\right)}, \quad (10)$$

where $\mathbf{v}_r \in \mathbb{R}^{h \times 1}$, $\mathbf{W}_r \in \mathbb{R}^{h \times d}$, and $\hat{\mathbf{W}}_r \in \mathbb{R}^{h \times d}$ are model parameters.

At the same time, a user's reviews are heterogeneous concerning a variety of items that the user has purchased, and not all reviews are relevant to the target item. Thus, similar to Eq. (6) and Eq. (7), given a user–item pair, a review-level co-attentive network is designed to select reviews from the user as guided by the reviews of the item.

Specifically, an affinity matrix at the review level is first computed by

$$\mathbf{G} = \phi(f([\mathbf{r}_1^u; ...; \mathbf{r}_n^u])^\top \mathbf{M}_r f([\mathbf{r}_1^v; ...; \mathbf{r}_m^v])), \quad (11)$$

where $\mathbf{M}_r \in \mathbb{R}^{d_r \times d_r}$ is a learnable parameter. Here, the (p, q)-th entry of $\mathbf{G}$ represents the affinity between the p-th review of the user and the q-th review of the item.

Then, the attention weights for the reviews of the user are calculated by

$$\boldsymbol{\beta}^u = \text{softmax}(\max_{\text{row}}(\mathbf{G} \otimes_{\text{row}} \boldsymbol{\beta}^v)), \quad (12)$$

where $\boldsymbol{\beta}^v = [\beta_1^v, ..., \beta_m^v]$ was obtained by Eq. (10) for the item. It is introduced to adapt $\mathbf{G}$ to encode important reviews of the item. Finally, we aggregate the review embeddings to represent a user by the following weighted sum.

$$\tilde{\mathbf{u}} = \sum_{i=1}^{n} \beta_i^u \mathbf{r}_i^u \quad (13)$$

Encoding Latent Rating Patterns. Although the embeddings $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ contain rich semantic information from reviews, there are some latent characteristics of users (items) that are not encoded by their reviews, but can be inferred from the rating patterns. For instance, a picky user might tend to uniformly pick lower ratings than a more easygoing user. To encode such personalized preferences, as inspired by (Koren, Bell, and Volinsky 2009), we embed a one-hot representation of the ID of each user (item) using an MLP, and obtain an embedding vector $\hat{\mathbf{u}}$ ($\hat{\mathbf{v}}$) for the user (item). This vector directly correlates with the ratings of a user (item), and is thus able to capture the latent rating patterns. Then, as illustrated in Fig. 2, we concatenate $\tilde{\mathbf{u}}$ and $\hat{\mathbf{u}}$ to obtain the final embedding of a user, i.e., $\mathbf{u} = [\tilde{\mathbf{u}}; \hat{\mathbf{u}}]$, and concatenate $\tilde{\mathbf{v}}$ and $\hat{\mathbf{v}}$ to obtain the final embedding of an item, i.e., $\mathbf{v} = [\tilde{\mathbf{v}}; \hat{\mathbf{v}}]$.

Prediction Layer

As shown by the top part of Fig. 2, the prediction layer receives $\mathbf{u}$ and $\mathbf{v}$, and concatenates them to $[\mathbf{u}; \mathbf{v}]$, which is then fed into a function $g(\cdot)$ to predict the rating. In this work, we realize $g(\cdot)$ as a parameterized factorization machine (FM) (Rendle 2010), which is effective to model the pairwise interactions between the input features for improving recommendation performance. Given an input $\mathbf{x} \in \mathbb{R}^{d\times 1}$, $g(\cdot)$ is defined as

$$g(\mathbf{x}) = b + \sum_{i=1}^{d} \mathbf{w}_i \mathbf{x}_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle \mathbf{z}_i, \mathbf{z}_j \rangle \, \mathbf{x}_i \mathbf{x}_j, \quad (14)$$

where b is a bias term, $\mathbf{w}$ is a parameter for linear regression, $\{\mathbf{z}_i\}_{i=1}^{d}$ are the factorized parameter for modeling the pairwise interactions between $\mathbf{x}_i$ and $\mathbf{x}_j$, $\langle \cdot, \cdot \rangle$ denotes the inner product, and the output of $g(\mathbf{x})$ is the predicted rating.

To learn model parameters, we minimize the difference between the true ratings and the predicted ratings, as measured by the mean squared error

$$\ell = \frac{1}{c} \sum_{i=1}^{c} (y_i - g([\mathbf{u}; \mathbf{v}]))^2, \quad (15)$$

where c is the total number of user–item pairs in the training data, and y_i is the true rating of the i-th user–item pair. The ℓ in Eq. (15) serves as our loss function for model training.

4 Experiments

In this section, we evaluate our AHN model on several real datasets and compare it with state-of-the-art approaches.

Datasets

We conducted experiments on 10 different datasets, including 9 Amazon product review datasets for 9 different domains, and the large-scale Yelp challenge dataset[1] on restaurant reviews. Table 1 summarizes the domains and statistics for these datasets. Across all datasets, we follow the existing work (Seo et al. 2017; Tay, Luu, and Hui 2018) to perform preprocessing to ensure they are in a t-core fashion, i.e., the datasets only include users and items that have at least t reviews. In our experiments, we evaluate the two cases of $t = 5$ and $t = 10$. For the Yelp dataset, we follow (Seo et al. 2017) to focus on restaurants in the AZ metropolitan area. For each dataset, we randomly split the user–item pairs into 80% training set, 10% validation set, and 10% testing set. When learning the representations for users and items, we only use their reviews from the training set, and none from the validation and testing sets. This ensures a practical scenario where we cannot include any future reviews into a user's (item's) history for model training.

Table 1: Statistics of datasets

Dataset	#Users	#Items	#Reviews
Digital_Music (DM)	5,541	3,568	64,706
Office_Products (OP)	4,905	2,420	53,258
Health (HE)	38,609	18,534	346,355
Toys_and_Games (TG)	19,412	11,924	167,597
Kindle_Store (KS)	68,223	61,935	982,619
Pets_Supplies (PS)	19,856	8,510	157,836
Tools_and_Home (TH)	16,638	10,217	134,476
Videos_Games (VG)	24,303	10,672	231,780
Automotive (AM)	2,928	1,835	20,473
Yelp	88,370	33,902	1,332,447

Compared Methods

We compare our model with both conventional approaches and state-of-the-art approaches, including Factorization Machines (FM) (Rendle 2010), SVD (Koren, Bell, and Volinsky 2009), Probabilistic Matrix Factorization (PMF) (Mnih and Salakhutdinov 2008), Nonnegative Matrix Factorization (NMF) (Lee and Seung 2001), DeepCoNN (Zheng, Noroozi, and Yu 2017), D-ATT (Seo et al. 2017), MPCN (Tay, Luu, and Hui 2018), and HUITA (Wu et al. 2019).

Among these methods, FM, SVD, PMF, and NMF are rating-based collaborative filtering methods. DeepCoNN, D-ATT, MPCN, and HUITA are state-of-the-art methods that leverage the semantic information in reviews for improved performance. Specifically, DeepCoNN uses the same CNN module to learn user and item embeddings based on their reviews for recommendation. D-ATT extends DeepCoNN by adding a dual-attention layer at word-level before convolution. MPCN attends to informative reviews by several pointers. HUITA uses a symmetric hierarchical structure to infer user (item) embeddings using regular attention mechanisms. It is worth noting that all of the above review-based methods regard user reviews and item reviews as the same type of documents and process them in an identical way.

Finally, to gain further insights on some of the design choices of our AHN model, we compare AHN with its variants, which will be discussed later in the ablation analysis.

Experimental Settings

The parameters of the compared methods are selected based on their performance on the validation set. Specifically, for

[1] https://www.yelp.com/dataset/challenge

Table 2: MSE results of the compared methods on different 5-core datasets

Dataset	FM	PMF	NMF	SVD	DeepCoNN	D-ATT	MPCN	HUITA	AHN
Digital_Music (DM)	0.8498	0.8788	1.0491	1.0843	0.8754	0.8506	0.8396	0.8719	**0.8172**
Office_Products (OP)	0.7291	0.7807	0.9285	0.7906	0.7253	0.7124	0.7084	0.7082	**0.6825**
Health (HE)	1.1825	1.2076	1.4317	1.1508	1.0862	1.0915	1.0817	1.1207	**1.0743**
Toys_and_Games (TG)	0.8639	0.9192	1.1105	0.9188	0.8391	0.8364	0.8452	0.8969	**0.8220**
Kindle_Store (KS)	0.6469	0.6695	0.8032	0.7370	0.6514	0.6382	0.6577	0.6544	**0.6270**
Pets_Supplies (PS)	1.3303	1.434	1.6806	1.3191	1.2598	1.2730	1.2566	1.3038	**1.2515**
Tools_and_Home (TH)	1.0229	1.1182	1.3580	1.0373	0.9856	0.9850	0.9871	1.0189	**0.9671**
Videos_Games (VG)	1.1849	1.2473	1.4357	1.3168	1.1575	1.1448	1.1747	1.1772	**1.1138**
Automotive (AM)	0.8189	0.9187	1.2074	0.8140	0.7809	0.7654	0.7643	0.7766	**0.7314**
Yelp	1.6094	1.8207	1.8389	1.6615	1.5957	1.5959	1.6195	1.6105	**1.5735**

Table 3: MSE results of the compared methods on different 10-core datasets

Dataset	FM	PMF	NMF	SVD	DeepCoNN	D-ATT	MPCN	HUITA	AHN
Digital_Music (DM)	0.8611	0.8641	0.9491	0.8503	0.8734	0.8429	0.8629	0.8512	**0.7880**
Office_Products (OP)	0.6291	0.6695	0.7346	0.6757	0.6016	0.5914	0.6120	0.6009	**0.5717**
Health (HE)	0.8166	0.9158	0.9200	0.8275	0.8328	0.8019	0.8020	0.8177	**0.7802**
Toy_and_Games (TG)	0.6904	0.6233	0.7575	0.6331	0.6331	0.6292	0.6412	0.6303	**0.5964**
Kindle_Store (KS)	0.5954	0.6035	0.6305	0.6483	0.5325	0.5275	0.5124	0.5312	**0.5092**
Pet_Supplies (PS)	1.2236	1.5239	1.2536	0.9950	0.9927	0.9616	1.0722	1.0168	**0.9421**
Tools_and_Home (TH)	0.8746	0.7668	0.9032	0.7391	0.6632	0.6297	0.6507	0.6445	**0.5948**
Videos_Games (VG)	1.0611	1.0718	1.2435	1.0318	1.0743	1.0365	1.0730	1.0697	**0.9927**
Yelp	1.5432	1.4734	1.5735	1.4025	1.3961	1.4018	1.4033	1.4040	**1.3671**

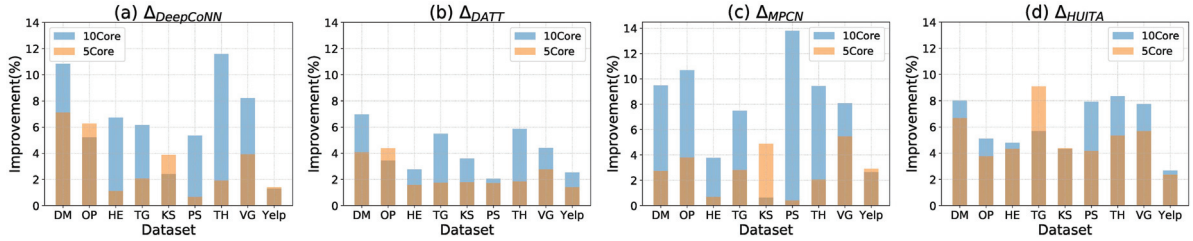

Figure 3: The relative improvements of AHN over (a) DeepCoNN, (b) D-ATT, (c) MPCN, and (d) HUITA, on different datasets. The abbreviations of the datasets can be found in Table 1-3. Here, blue refers to the improvement on the 10-core datasets, orange refers to the improvement on the 5-core datasets, brown is the overlapped area between blue and orange.

FM, the dimensionality of the factorized parameters is 10. For SVD, PMF, and NMF, the number of factors is set to 50. DeepCoNN uses 100 convolutional kernels with window size 3. D-ATT uses 200 filters and window size 5 for local attention; 100 filters and window sizes [2, 3, 4] for global attention. MPCN uses 3 pointers, and hidden dimensionality of 300 for inferring affinity matrix. HUITA uses 200 filters in the word-level CNN with window size 3, and 100 filters in the sentence-level CNN with window size 3.

For our AHN model, the dimensionality of the hidden states of the BiLSTM is set to 150. The dimensionality of the user and item ID embeddings are set to 300. The dimensionality of $\mathbf{M}_s$ ($\mathbf{M}_r$) in Eq. (6) (Eq. (11)) is 300. We apply dropout (Srivastava et al. 2014) with rate 0.5 after the fully connected layer to alleviate the overfitting problem. The loss function is optimized by Adam (Kingma and Ba 2014), with a learning rate of 0.0002 and a maximum of 10 epochs.

For the methods DeepCoNN, D-ATT, and HUITA, the pre-trained GloVe (Pennington, Socher, and Manning 2014) is used to initialize the word embeddings. For MPCN and our AHN, the word embeddings are learned from scratch since using pre-trained embeddings generally degrades their performance. For all methods, the dimensionality of the word embedding is set to 300. We independently repeat each experiment 5 times, and use the averaged mean square error (MSE) (Zheng, Noroozi, and Yu 2017) to quantitatively evaluate the performance.

Experimental Results

Table 2 summarizes the results of the compared approaches on the 5-core datasets. We have several observations from the results. First, review-based methods generally outper-

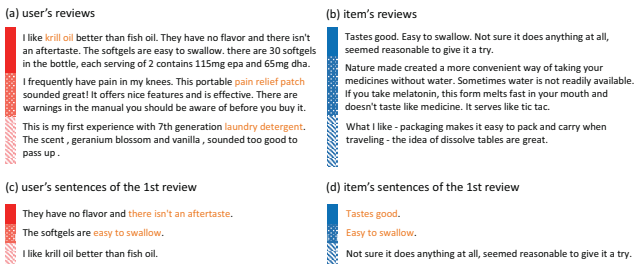

Figure 4: The visualization of attention weights on (a) user's reviews, (b) item's reviews, (c) user's sentences, (d) item's sentences. The item is a sleep aid medicine. The vertical bars represent weights. Darker colors indicate higher weights.

form rating-based methods. This validates the usefulness of reviews in providing fine-grained information for refining user and item embeddings for improving the accuracy of rating prediction. Second, methods that distinguish reviews, such as D-ATT and MPCN, often outperform DeepCoNN, which suggests that different reviews exhibit different degrees of importance for modeling users and items. We also observe that HUITA does not show superiority over DeepCoNN. This may stem from its symmetric style of attention learning, which does not make much sense when reviews are heterogeneous. Finally, the proposed AHN consistently outperforms other methods, which demonstrates the effectiveness of distinguishing the learning of user and item embeddings via asymmetric attentive modules so as to infer more reasonable attention weights for recommendation.

Table 3 presents the results on the 10-core datasets, from which the *Automotive* dataset is excluded because only very few users and items are left after applying the 10-core criterion on it. In contrast to Table 2, all methods in general achieve better results in Table 3, since more ratings and reviews become available for each user and item. In this case, we observe that D-ATT often outperforms MPCN. This may be because the Gumbel-Softmax pointers in MPCN make hard selections on reviews, thereby filtering out many reviews that may result in a significant loss of information. This problem is more severe when users (items) have more useful reviews, as in the 10-core scenario. Additionally, we observe that the performance gaps between AHN and the compared methods become larger. To see it, we summarize the relative improvements of AHN over each of the review-based methods in Fig. 3. From the figure, we observe that AHN generally gains more on the 10-core datasets, with absolute gains of up to 11.6% (DeepCoNN), 7.0% (D-ATT), 13.8% (MPCN), and 8.4% (HUITA). This suggests that the more reviews each user and item has, the more important it is to perform proper attention learning on relevant reviews and sentences at the users' side and item's side, respectively.

Case Study

In this section, we investigate the interpretability of AHN. Fig. 4(a) and (b) show the attention weights of AHN on the top three reviews of a pair of user and item on the *Health* dataset, where the item is a sleep aid medicine. In each of the user's reviews, the highlighted words indicate the item described by the review. As can be seen, the first two items "krill oil" and "pain relief patch" are more relevant to the item "sleep aid medicine" than the "laundry detergent" in the lowest-weighted review. On the other hand, the top two reviews of the item are more informative with regard to the aspects of the item than the last review, which only describes about "packaging", a marginal aspect of a medicine. Thus, the review-level attention weights of AHN are meaningful.

Fig. 4(c) and (d) zoom into the attention weights of AHN on the top three sentences of the first review of the user and item, respectively. The highlighted words indicate the reason of why the sentences are ranked highly. As can be seen, the user cares about the taste of a medicine and prefers easily-swallowed softgels, while the item indeed appears to taste good and is easy to swallow. It is worth noting that although the first two sentences in Fig. 4(d) are short, they convey more useful information that the lowest-weighted sentence. Thus, the sentence-level attention weights of AHN are also meaningful. This explains why AHN predicts a 4.4 rating score on this user–item pair, which is close to the true rating 5.0 as marked by the user.

Table 4: Ablation analysis

Model	VG	DM	AM	OP
AHN	**1.1138**	**0.8172**	**0.7314**	**0.6825**
(a) –Item aggregators	1.1286	0.8205	0.7506	0.6951
(b) –User aggregators	1.1604	0.8246	0.7467	0.6941
(c) –Adapted affinity	1.1363	0.8229	0.7348	0.6936
(d) –FM	1.1267	0.8341	0.7723	0.7078
(e) –Gating	1.1220	0.8188	0.7385	0.6883

Ablation Analysis

Table 4 presents the results of our ablation analysis using four datasets. In the table, AHN is our original model. In (a), the item's attention modules are replaced by average-pooling. In (b), the user's co-attention modules are replaced by attention modules similar to the item's ones and thus constitutes a symmetric model. In (c), we remove the row-wise multiplication between the affinity matrix and the attention weights in Eq. (7) and Eq. (12). In (d), the parameterized factorization machine is replaced by dot product. In (e), the gating mechanisms in Eq. (5) and Eq. (10) are removed.

From Table 4, we observe that different variants of AHN show suboptimal results to various degrees. Comparing with (a), we can observe the importance of considering attention weights on the sentences and reviews of each item. The degraded MSEs of (b) suggest that our asymmetric design in the model architecture is essential. The results of (c) validates our design of the attention-adapted affinity matrix in Eq. (7) and (12). The substantial MSE drops in (d) signify the superiority of using FM as the prediction layer. The comparison between (e) and AHN suggests the effectiveness of the gating mechanisms. Thus, the results of the ablation study validate the design choices of our model architecture.

5 Conclusions

In this work, we highlight the asymmetric attention problem for review-based recommendation, which has been ignored by existing approaches. To address it, we propose a flexible neural architecture, AHN, which is characterized by its asymmetric attentive modules for distinguishing the learning of user embeddings and item embeddings from reviews, as well as by its hierarchical paradigm to extract fine-grained signals from sentences and reviews. Extensive experimental results on datasets from different domains demonstrate the effectiveness and interpretability of our method.

References

Bao, Y.; Fang, H.; and Zhang, J. 2014. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *Twenty-Eighth AAAI conference on artificial intelligence (AAAI)*.

Catherine, R., and Cohen, W. 2017. Transnets: Learning to transform for recommendation. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, 288–296. ACM.

Chen, C.; Zhang, M.; Liu, Y.; and Ma, S. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the World Wide Web Conference (WWW)*, 1583–1592. International World Wide Web Conferences Steering Committee.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, X., and De Melo, G. 2018. A helping hand: Transfer learning for deep sentiment analysis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2524–2534.

Ilse, M.; Tomczak, J.; and Welling, M. 2018. Attention-based deep multiple instance learning. In *International Conference on Machine Learning (ICML)*, 2132–2141.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* 42:30–37.

Lee, D. D., and Seung, H. S. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems (NIPS)*, 556–562.

Ling, G.; Lyu, M. R.; and King, I. 2014. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, 105–112. ACM.

McAuley, J., and Leskovec, J. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, 165–172. ACM.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*, 3111–3119.

Mnih, A., and Salakhutdinov, R. R. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems (NIPS)*, 1257–1264.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *NAACL-HLT*, 2227–2237.

Rendle, S. 2010. Factorization machines. In *International Conference on Data Mining (ICDM)*, 995–1000. IEEE.

Santos, C. d.; Tan, M.; Xiang, B.; and Zhou, B. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Seo, S.; Huang, J.; Yang, H.; and Liu, Y. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, 297–305. ACM.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958.

Tay, Y.; Luu, A. T.; and Hui, S. C. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2309–2318. ACM.

Wang, S.; Yu, M.; Chang, S.; and Jiang, J. 2018. A co-matching model for multi-choice reading comprehension. *arXiv preprint arXiv:1806.04068*.

Wu, C.; Wu, F.; Liu, J.; and Huang, Y. 2019. Hierarchical user and item representation with three-tier attention for recommendation. In *NAACL-HLT*, 1818–1826.

Xiong, C.; Zhong, V.; and Socher, R. 2017. Dynamic coattention networks for question answering. In *ICLR*.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhang, X.; Li, S.; Sha, L.; and Wang, H. 2017. Attentive interactive neural networks for answer selection in community question answering. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 425–434. ACM.