Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction

Sungyong Seo
University of Southern California
sungyons@usc.edu

Jing Huang
Visa Research, Visa Inc.
jinhuang@visa.com

Hao Yang
Visa Research, Visa Inc.
haoyang@visa.com

Yan Liu
University of Southern California
yanliu.cs@usc.edu

ABSTRACT

Recently, many e-commerce websites have encouraged their users to rate shopping items and write review texts. This review information has been very useful for understanding user preferences and item properties, as well as enhancing the capability to make personalized recommendations of these websites. In this paper, we propose to model user preferences and item properties using convolutional neural networks (CNNs) with dual local and global attention, motivated by the superiority of CNNs to extract complex features. By using aggregated review texts from a user and aggregated review text for an item, our model can learn the unique features (embedding) of each user and each item. These features are then used to predict ratings. We train these user and item networks jointly which enable the interaction between users and items in a similar way as matrix factorization. The local attention provides us insight on a user's preferences or an item's properties. The global attention helps CNNs focus on the semantic meaning of the whole review text. Thus, the combined local and global attentions enable an interpretable and better-learned representation of users and items. We validate the proposed models by testing on popular review datasets in *Yelp* and *Amazon* and compare the results with matrix factorization (MF), the hidden factor and topical (HFT) model, and the recently proposed convolutional matrix factorization (ConvMF+). Our proposed CNNs with dual attention model outperforms HFT and ConvMF+ in terms of mean square errors (MSE). In addition, we compare the user/item embeddings learned from these models for classification and recommendation. These results also confirm the superior quality of user/item embeddings learned from our model.

KEYWORDS

Convolutional Neural Network; Attention Model; Deep Learning for Recommender Systems

RecSys'17, August 27–31, 2017, Como, Italy.

ISBN 978-1-4503-4652-8/17/08…$15.00.
DOI: http://dx.doi.org/10.1145/3109859.3109890

1 INTRODUCTION

Recommender systems are ubiquitous today at online shopping websites such as Amazon and Netflix. Ever since the famous Netflix Prize competition started ten years ago, collaborative filtering (CF) techniques have become the successful and dominant approaches for recommender systems. Many of the CF approaches are based on matrix factorization (MF) [11], which decomposes the user-item rating matrix into two matrices corresponding to latent features of users and items. The dot product between a user and an item feature vector is then used to predict the rating that the user would assign to the item.

Collaborative filtering has its own limitations and drawbacks, however. First, it is difficult to produce reliable recommendations for users with few ratings or recommend items with few ratings (the well-known cold start problem). Popular items tend to receive more recommendations, while new items are left with no chance to be exposed. Another drawback of CF/MF techniques is content-ignorance. In other words, contextual information such as properties of items or profiles of users is not considered for recommendations. Since the additional information can provide details about particular user or item, it is difficult to build personalized recommender systems without fully utilizing these data.

Using review text is one approach to alleviate the above issues. Most shopping websites encourage their users to rate shopping items and write reviews. Review text complements the rating numbers by providing rich information of items and implicit preferences of users. Review text explains why a user assigns such a rating to an item. A set of all reviews from this user allows us to derive this user's preferences; similarly, a set of reviews for an item describes prominent properties of the item rated by many users. These preferences of a user and properties of an item can then be leveraged to alleviate the cold-start problem when the ratings are few, and present explainable recommendation results to users [30]. Recently using review text information has shown to improve rating prediction accuracy, especially for users and items with few ratings [8, 14, 17].

The RMR model in [14] uses interpretable latent topic models like LDA [4] on item review text: an item document consists of all reviews of an item, and the latent topic distribution is derived from these item documents. The topics discovered from item documents may not be suitable for users. It is believed that the aggregated review text of items cannot cover the same sentimental expressions for each individual user. For example, the same word "nice" might indicate different sentimental meaning from different users, while

one user links "nice" to a rating of 4, another would give it a rating of 5. Therefore, we should not discard user-specific meaning of words and ignore their effect on the ratings by just assuming the topic distribution is the same as those in the item review. Each user has their own preferences or tastes that can be discovered using all reviews written by this user. Furthermore, a topic model is specifically designed for discovering latent semantic structures in a text body and the aspect of a topic model limits generalities of extracted features.

Recently, CNNs have shown great success in many natural language processing tasks, such as text classification, sentiment analysis, neural language model and information retrieval ([9, 22, 23, 29]). From these works, it is proved that CNNs are able to extract more general contextual features from texts and the features have been successfully used to build recommender systems showing less prediction error than that of CF/MF ([6, 8]). However, it is believed that features from neural networks are not interpretable in general. Although many of the learned filters in the first convolutional layer capture features similar to *n*-grams when word embeddings input to the first convolutional layer, it becomes much more difficult to understand features come out of few more convolutional layers.

Therefore, in this paper we propose to model the user and the item separately: the aggregated review text from a user is used to build a user-specific model and the aggregated review text on an item is used to build an item-specific model. We use convolutional neural networks (CNNs) to extract embedded representations of users and items from their corresponding review text.

Furthermore, we put an attention layer after the word embedding layer and before the convolutional layer. This attention layer learns what is important from a local or global window of words by learning weights for these words. The idea of using both local and global attention modules with CNNs is inspired by the work in [15] where a local and global attention were experimented for neural machine translation. In our work, the local attention selects informative keywords from a local window before the words are fed into the convolutional layer, and the global attention aims to ignore noisy and irrelevant words from long reviews and captures the global semantic meaning. Both attention layers give us the ability to interpret and visualize what the model is doing: for example, words with higher (or lower) weights are highlighted in Table 4 and Table 5.

The main contributions of this paper are summarized as follows:

- Interpretable attention-based CNNs are used to learn user and item representations from corresponding user and item reviews; these representations are used to predict the ratings of a user for an item just like the MF technique. To the best of our knowledge, this is the first paper to use attention-based CNNs for rating prediction.
- Dual attention layers are used in our network: a local layer and a global layer. The attention layer is used before the convolutional layers to select informative words from a local or global window that contributes to the rating. These attention layers give us insight to the words that are selected by the models that highlight a user's preferences or an item's properties.
- Our model outperforms the baseline MF, HFT, and the recently proposed model, ConvMF+ on the *Yelp* Challenge dataset 2013 and the *Amazon* datasets. In addition, the representations (embeddings) learned for user/item from our model are shown to be better than those from HFT and ConvMF+ for other application purposes, such as item classification and recommendation ranked list for a query user.

The rest of this paper is organized as follows: related works are first reviewed in Section 2. Section 3 describes the components of our networks in detail . Experimental setup and result analysis are presented in Section 4 and Section 5; and finally Section 6 concludes with future work.

2 RELATED WORK

There are two lines of research related to our work: the first uses review text for recommendation, and the second is recent research on sentiment analysis using deep learning. We present brief reviews for these two research areas as follows.

Recent works on using review text for recommendations focus mostly on topic modeling from review text, such as hidden factors as topics (HFT) [17], ratings meet reviews (RMR) [14] and TopicMF [3]. HFT employs a LDA-like topic model on review text for users and items, and a matrix factorization (MF) model to fit the ratings. The two models are combined in an objective function that uses the likelihood of the review text modeling by the topic distribution as a regularization term for the latent user-item parameters. This approach is shown to improve significantly over the baselines that use ratings or reviews alone, and it also works with items with only a few reviews. RMR [14] shares the same LDA-like topic modeling on item reviews as HFT, except that RMR uses Gaussian mixtures to model the rating instead of MF-like techniques. The mixture weights are assumed to have the same distribution as the topic distribution. TopicMF [3], jointly models user ratings with MF and review text with non-negative matrix factorization (NMF) to derive topics from the reviews. A transform function is used to align the topic distribution parameters with the corresponding latent user and item factors. HFT learns the topics for each item, while topicMF learns the topics for each review. Their experimental results show topicMF improved upon HFT. However, the exponential function used as the transformation function might limit the flexibility of topic distributions. As pointed out in [3], the authors hope to model the user's preferences directly to their rating behavior in their future work, which is what our model is designed to do.

Recently, deep learning techniques have been applied to recommender systems with review content. In [1], the proposed model consists of a MF model for learning the latent factors and a recurrent neural network (RNN) for modeling the likelihood of the review using an item's latent factors. The RNN model is combined with MF via a regularization term, just like the approach used in HFT. In [26] a hierarchical Bayesian model called collaborative deep learning (CDL) is proposed to take advantage of reviews. CDL uses stacked denoising autoencoders (SDAE) to learn latent feature representations for the items. This network together with collaborative filtering for the rating matrix are jointly trained. However, the

content information is only extracted from bag-of-words representations, which does not take into account word orders and context that are important for extracting semantic meaning. To improve upon the bag-of-words representation, convolutional neural networks are integrated into MF in [8] in order to capture contextual information in item reviews for the rating prediction. The proposed ConvMF+ is shown to outperform PMF and CDL. All the above works fail to explicitly link the user's preferences and sentiment in their review text to their ratings.

Another line of research closely related to recommendation on review text is sentiment analysis and text classification. Recently, there are many works on sentiment analysis and text classification using deep learning techniques that achieve impressive results as shown in [5, 7, 12, 22, 24, 25, 28, 29]. Various deep neural network configurations are used for these tasks: CNN [5, 7, 22, 25, 29], recursive neural tensor network, recurrent neural network [12, 24] and LSTM [28]. Most networks use word embeddings [12, 22, 24, 25, 28] as the input layer, while [5, 7, 29] used character embeddings, especially [5] uses very deep (29 layers) CNNs on character embeddings and shows impressive results on text classification, including the Amazon reviews.

The idea of attention in neural networks is loosely based on the visual attention mechanism found in humans. Attention-based neural networks have been applied successfully to various tasks, such as neural machine translation [2, 15], image caption generation [27] and document classification [28]. A big advantage of attention is that it gives us the ability to interpret and visualize what the model is doing. For example, in [27], a CNN is used to "encode" an image, and a recurrent neural network with attention is used to generate a caption. By visualizing the attention weights, we interpret which part of the image the model is looking at while generating a description word. While attention-based deep learning models rely on RNNs and encoder-decoders for tasks such as machine translation and image caption generation, the attention module in our model is designed to work with CNNs. It allows us to infer informative words that are linked directly to the user's rating. We put attention layers before the CNNs so that we can visualize those informative words at the end of model training and help us interpret the results (as shown in Table 4,5).

3 PROPOSED MODEL

In this section, we describe our dual attention-based model, D-Attn, for learning latent representations from review text. Figure 1 shows the overall architecture. We use the same network structure for the user and the item network. So, we describe the user network in detail: the left part is the local attention-based module (L-Attn) that learns representations of local informative keywords. The right part is the global attention-based module (G-Attn) that learns representations from the original review word sequences. These two representations are then concatenated and passed through fully-connected (FC) layers as the final learned representation for the user (and item) rating prediction.

We describe each of these modules in Figure 1. We denote scalars with italic lower-cases (x, y), vectors with bold lower-cases ($\mathbf{x}, \mathbf{z}$), and matrices or high dimensional tensors with bold upper-cases ($\mathbf{X}, \mathbf{W}$). In addition, we use MATLAB-like array indexing, thus, $\mathbf{W}(:, j)$ means elements of j-th column of the matrix $\mathbf{W}$.

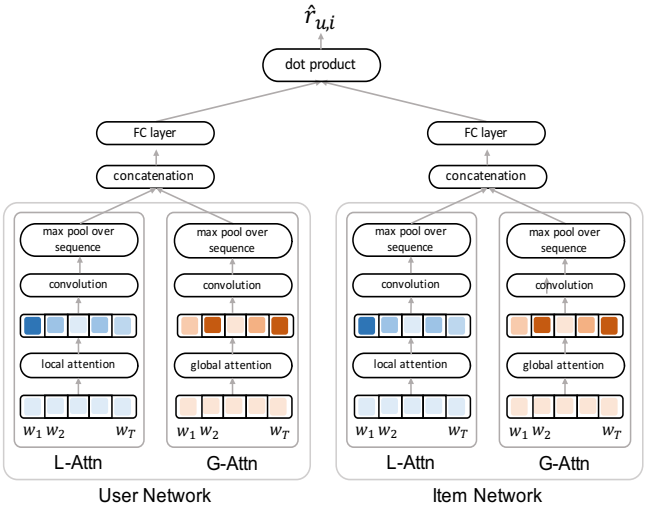


Figure 1: Architecture of D-Attn to extract latent representations of users and items. A user document D_u and an item document D_i are fed into (Left) the user network and (Right) the item network, respectively.

Embedding layer We use word embeddings for the input review document D_u: a set of reviews from user u. An embedding layer can be simply regarded as a look-up operation that reads a one-hot vector, $\mathbf{e}_t \in \mathbb{R}^{|\mathcal{V}|}$, for a word as an input, and maps it to a dense vector, $\mathbf{x}_t = (x_1, x_2, \cdots, x_d)$ and $\mathbf{x}_t \in \mathbb{R}^d$. The weight of the embedding layer is $\mathbf{W}_e \in \mathbb{R}^{d \times |\mathcal{V}|}$:

$$\mathbf{x}_t = \mathbf{W}_e \mathbf{e}_t. \tag{1}$$

and $\mathcal{V}$ is a set of words. $|\mathcal{V}|$ is the size of the vocabulary.

Local attention (L-Attn) module on the left In our model, the local attention module, L-Attn, is used to learn which words are more informative in a local window of words. Let D_u be represented as a length T word embeddings $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$. Then, we apply the attention through sliding kernels to this sequence. Let $\mathbf{x}_i$ be the center word and the w be the kernel width. We compute weighting scores for each word with a $\mathbf{W}^1_{l-att} \in \mathbb{R}^{w \times d}$ parameter matrix and a bias b^1_{l-att} as follows:

$$\mathbf{X}_{l-att,i} = (\mathbf{x}_{i+\frac{-w+1}{2}}, \mathbf{x}_{i+\frac{-w+3}{2}}, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_{i+\frac{w-1}{2}})^\top, \tag{2}$$

$$\mathbf{s}(i) = g(\mathbf{X}_{l-att,i} * \mathbf{W}^1_{l-att} + b^1_{l-att}), \qquad i \in [1, T]. \tag{3}$$

where $*$ is an operation which means element wise multiplication and sum. We use the sigmoid function (σ) for the activation function, $g(\cdot)$.

$\mathbf{s}(i)$ are the attention scores used as a weight for i-th word embedding. Hence, if a given word has a smaller score than another word, we can interpret that the smaller score word is less important than the larger score word in our model.

$$\hat{\mathbf{x}}^L_t = \mathbf{s}(t)\mathbf{x}_t. \tag{4}$$

$\hat{\mathbf{x}}_t^L$ where $t \in [1, T]$ are a weighted sequence of word embeddings.

The weighted sequence of word embeddings are then passed through a convolutional layer with a kernel size of 1 (the 1x1 convolution was first introduced in [13]). More kernel sizes bigger than 1 could be used, but since the local attention is designed to extract individual preference or property keywords for user/item, the kernel size 1 fits exactly this purpose, and it suffers less over-fitting. A fixed length summary vector is then extracted by applying max pooling over the sequence. As a result, *local attention representation* of given text is obtained by following convolution with a matrix $\mathbf{W}_{l-att}^2 \in \mathbb{R}^{d \times n_{l-att}}$ and a bias $\mathbf{b}_{l-att}^2 \in \mathbb{R}^{n_{l-att}}$ and max pooling:

$$\mathbf{Z}_{l-att}(t,i) = g(\hat{\mathbf{x}}_t^L * \mathbf{W}_{l-att}^2(:,i) + \mathbf{b}_{l-att}^2(i)), \quad i \in [1, n_{l-att}] \tag{5}$$

$$\mathbf{z}_{l-att}(i) = \textsc{Max}(\mathbf{Z}_{l-att}(:,i)). \tag{6}$$

n_{l-att} is the number of filters and $g(\cdot)$ is a tanh function.

Global attention (G-Attn) module on the right The sequence of words (with its original order) from D_u is an input to the global attention module, G-Attn. The input passes through the global attention layer by multiplying the global attention scores with themselves. This process is similar to that of L-Attn, however, the attention scores are computed from the entire input text. Let $\hat{\mathbf{x}}_t^G$ be a weighted word embedding by the G-Attn module where $t \in [1, T]$. By using the global attention layer, the effects of uninformative words are diminished and the global semantic meaning is captured more precisely through the following CNN layer. For the convolutional layer, we set the length of a filter as w_f, which means the filter operates on w_f words. If the number of filters is n_{g-att}, the convolution filters $\mathbf{W}_{g-att} \in \mathbb{R}^{w_f \times d \times n_{g-att}}$ are applied to a sequence of w_f weighted word embeddings, $\hat{\mathbf{X}}_{g-att,i} \in \mathbb{R}^{w_f \times d}$, and output features $\mathbf{Z}_{g-att} \in \mathbb{R}^{(T-w_f+1) \times n_{g-att}}$:

$$\hat{\mathbf{X}}_{g-att,i} = (\hat{\mathbf{x}}_i^G, \hat{\mathbf{x}}_{i+1}^G, \cdots, \hat{\mathbf{x}}_{i+w_f-1}^G)^\top, \tag{7}$$

$$\mathbf{Z}_{g-att}(i,j) = g(\hat{\mathbf{X}}_{g-att,i} * \mathbf{W}_{g-att}(:,:,j) + \mathbf{b}_{g-att}(j)), \quad i \in [1, T-w_f+1], \quad j \in [1, n_{g-att}]. \tag{8}$$

$g(\cdot)$ is a tanh function and $\mathbf{b}_{g-att}$ is a bias vector. In the pooling layer, we choose heuristically a max pooling over the sequence: $\mathbf{z}_{g-att}(j) = \textsc{Max}\left(\mathbf{Z}_{g-att}(:,j)\right)$. We can obtain as many $\mathbf{z}_{g-att}$ as the number of different filter length w_f. In this work, n_w different filter lengths are used.

Final layers The outputs of the local and global attention module are concatenated, and run through additional fully connected layers $\mathbf{W}_{FC}^1$ and $\mathbf{W}_{FC}^2$:

$$\mathbf{z}_{out}^1 = \mathbf{z}_{l-att} \oplus \mathbf{z}_{g-att}^1 \oplus \cdots \oplus \mathbf{z}_{g-att}^{n_w}, \tag{9}$$

$$\mathbf{z}_{out}^2 = g(\mathbf{W}_{FC}^1 \cdot \mathbf{z}_{out}^1 + \mathbf{b}_{FC}^1), \tag{10}$$

$$\gamma_u = g(\mathbf{W}_{FC}^2 \cdot \mathbf{z}_{out}^2 + \mathbf{b}_{FC}^2). \tag{11}$$

$\oplus$ is a concatenation operator. $g(\cdot)$ is a ReLU function.

Training of the network Two different channels (L-Attn and G-Attn) are used to learn two different latent representations, *local attention representation* and *global semantic representation*. These two are then merged into one attention-based semantic representation. Let γ_u be the attention-based semantic vector from the users' network and γ_i be the corresponding vector from the items' network. As in CF/MF techniques, the latent representations (γ_u, γ_i) are mapped into the same vector space $(\mathbb{R}^K)$ and the ratings can be estimated by the inner product.

$$\hat{r}_{u,i} = \gamma_u^\top \gamma_i \tag{12}$$

The estimation can be considered as a regression problem and all parameters in the two networks (user network and item network) are trained jointly through the backpropagation technique. Mean Squared Error (MSE) is used as a loss function. In training time, D_u and D_i are fed into two networks respectively. At test time, a pair of a user and an item (u, i) along with their corresponding D_u and D_i are fed through the user/item networks, and the inner product of γ_u, γ_i is the estimated rating $\hat{r}_{u,i}$.

4 EXPERIMENTAL SETUP

We evaluate our proposed models using open datasets from *Yelp* and *Amazon*. In this section, we describe these datasets as well as our experimental setup.

4.1 Datasets

We use two publicly available datasets that provide user reviews and rating information. The first dataset is from the *Yelp Business Rating Prediction Challenge* 2013[1], which includes reviews on restaurants in the Phoenix, AZ metropolitan area. The second dataset is the *Amazon Product Data*[2], which contains millions of product reviews and metadata from *Amazon*. This dataset has been investigated by many researchers [17–19]. In this paper, we focus on training interpretable models rather than dealing with the cold start problem. Therefore, we start our experiments with the *5-core* subset, which has at least 5 reviews for each user or item. We use six product categories in this work. The key characteristics of these datasets are summarized in Table 1.

Table 1: Statistics of Dataset

Dataset	*Yelp*	*Amazon*
# of Users	45,980	30,759
# of Items	11,537	16,515
# of Reviews	229,900	285,644
Avg. # of Words per a review	130	104
Avg. # of Reviews per a user	5.00	9.29
Avg. # of Reviews per an item	17.30	19.93

We randomly divided each dataset into training, validation and test sets (80%, 10%, 10% respectively).

4.2 Data Preprocessing

The first step in our data processing pipeline is to remove non-contextual words such as html tags, url, or symbols. Next, we concatenate all reviews from the same user, say user u, into a document D_u. Similarly, we concatenate all reviews on the same item, say item

[1] https://www.kaggle.com/c/yelp-recsys-2013
[2] http://jmcauley.ucsd.edu/data/amazon/

i, into a document D_i. We put a delimiter between two different reviews from one user (or item) because if two reviews are concatenated without a delimiter, convolution filters will mix words which are not contextually connected. As expected, the length of these concatenated review documents follows a long-tailed distribution and we set the length of each document to conserve at least 70% of the words. The resulting documents are fed into the embedding layer as described in Section 3.

4.3 Baselines

We implemented several comparable baselines as comparisons to our proposed model. The first one is Matrix Factorization (MF)[3] [11] that characterizes both users and items by vectors of factors inferred from item rating patterns. The second baseline is Hidden Factors as Topics (HFT) [17]. We set the number of hidden topics K as 5 which is reported in [17]. We also compare our model to a recently proposed context-aware recommendation model, convolutional matrix factorization (ConvMF+) [8]. We use ConvMF+ which is based on the GloVe embedding to represent reviews as we do.

To see the effectiveness of the global attention layer, we report results from CNN-only which is a subpart of G-Attn. In addition, results from only one module, a local attention (L-Attn) or a global attention module (G-Attn), are also compared to show the synergy of the local and global attention modules. Finally, we show a naive method, Offset, which simply uses the average rating (μ) as the prediction. This baseline shows how ratings in a given dataset are biased and the upper bounds of rating estimates for each dataset.

4.4 Parameter Setting

We follow the guideline in [31] for setting initial parameters of CNN models. We use pre-trained word embedding GloVe [21] based on aggregated global word-word co-occurrence statistics from a corpus. The dimension of embedding is $d = 100$ and the most frequent 30,000 words are used. In the L-Attn module, we use $w = 5$ window size with a sigmoid function(σ) and $n_{l-att} = 200$ in $\mathbf{W}^2_{l-att}$.

In the G-Attn module, we use three different filter lengths $w_f \in [2, 3, 4]$ and $n_{g-att} = 100$ for each $\mathbf{W}_{g-att}$. These filter sizes correspond to n-gram features. Finally, to combine local and global attention representations, the fully connected layers $\mathbf{W}^1_{FC}$ and $\mathbf{W}^2_{FC}$ with the number of hidden factors, $K_1 = 500$ and $K_2 = 50$ are used, respectively. 0.5 dropout probabilities are used for the two FC layers to reduce overfitting. Activation functions of convolution layers in both L-Attn and G-Attn are tanh and ReLUs are used for the FC layers. Adam optimizer[10] is used with a 1e-4 learning rate. Our experiments are done with Theano running on GPU machines of Nvidia GeForce GTX TITAN X.

5 RESULTS AND DISCUSSION

5.1 Rating Estimation

The Mean Squared Error (MSE) of the rating estimations are shown in Table 2 for the D-Attn model as well as for other models. To obtain reliable results, we repeat our model training/testing on four random splits of the data and report the average prediction errors. The standard deviations of prediction errors are 0.004 for Yelp and 0.001 for Amazon. We can see that the D-Attn outperforms all baselines on the *Yelp* dataset and the *Amazon* datasets. This clearly confirms the effectiveness of our proposed method.

We further break down the results for the Amazon dataset by product category, as shown in Table 3. While MF and HFT each have their own strengths, our D-Attn model can achieve the best of both, which can be seen from the results on these datasets. First, the ratings in the Yelp dataset are not consistent. In other words, a user gives a broad range of ratings to different items in general. This explains why MF leads to a large MSE on this dataset. Second, the review text may also be inconsistent. On the *Automotive* dataset, HFT shows a worse result than that of MF. This is because each user is likely to use different words for different items, and each item has reviews written using a diverse set of words. As a result, it is difficult for HFT to infer proper topics for such heterogeneous corpus. Although ConvMF+ constantly shows lower MSE than those of MF and HFT, our model is even better than ConvMF+, because our model is designed to understand users' preferences and items' properties with both the local attention layer and the global attention layer in order to decide which words are more important for the rating estimation.

We also present the results of the subpart models: CNN-only, L-Attn-only, and G-Attn-only. Each model has its own limitations. First, the CNN-only model is able to capture contextual meaning by convolutional layers in the model. However, it treats every word equally in extracting contextual features. The G-Attn-only model which adds the global attention module before the CNN-only model can extract contextual features as well as diminish the influence of less important words. Thus, the results of the G-Attn-only model are better than those of the CNN-only model. Interestingly, the L-Attn-only model shows the best results among these three subpart models. This suggests that weighting words is essential and how the weights are computed makes a difference too.

The ConvMF+ model integrates two subparts: 1) Probabilistic Matrix Factorization (PMF) for combining latent user and item vectors and 2) CNN for item contextual modeling. On the other hand, our approach uses attention-based CNN models for both users and items. Next, we show visually what the local and global attention layers do, and the effectiveness of the user/item embeddings learned from our CNN models for the practical purpose of classification and recommendation.

5.2 Visualization of Attention Layers

We highlight words that have high scores computed by the local attention modules in Table 4. We randomly sample an identical review example from a user in the Yelp dataset. Words with the highest scores are colored dark-green, high scoring words are light-green, and low/medium score words are not colored. In Table 4, we compare attention scores trained by the L-Attn-only model with the D-Attn model, respectively.

We can see a few interesting patterns from these results. Overall, similar words are highlighted in the L-Attn-only model and the D-Attn model. However, the L-Attn-only model gives high scores to consecutive words including trivial pronoun words. For example, it highlights "*They are good people great atmosphere*" as high score words. On the other hand, the D-Attn model is more precise in that

[3] We use MyMediaLite package. http://www.mymedialite.net

Table 2: MSE of rating estimation for different models (the best results are starred)

Dataset	Offset	MF	HFT	ConvMF+	CNN-only	L-Attn-only	G-Attn-only	D-Attn
Yelp	1.484	1.275	1.224	1.209	1.210	1.203	1.205	1.191*
Amazon	1.011	0.876	0.871	0.858	0.864	0.860	0.861	0.855*

Table 3: Breakdown of *Amazon* dataset evaluation by product category

Dataset	Offset	MF	HFT	ConvMF+	CNN-only	L-Attn-only	G-Attn-only	D-Attn
Amazon instant video	1.256	0.968	0.957	0.950	0.952	0.950	0.951	0.946*
Automotive	0.856	0.786	0.802	0.780	0.785	0.781	0.782	0.779*
Grocery and gourmet food	1.202	1.017	1.004	1.002	1.005	1.003	1.004	0.997*
Musical instruments	0.730	0.725	0.714	0.698	0.704	0.700	0.701	0.694*
Office products	0.867	0.724	0.723	0.720	0.727	0.721	0.723	0.719*
Patio lawn and garden	1.153	1.037	1.025	1.001	1.008	1.002	1.003	0.996*

it successfully picks "*great*" from the sequence of words. Furthermore, many neutral words such as "*Arizona*", "*I've*", and "*that*" are incorrectly emphasized by the L-Attn-only windows. Hence, this example shows that the local attention modules work similarly in both models but D-Attn is more precise with the help of global attention.

Table 4: Highlighted words by local attention scores in a user's review. Top: attention scores in the L-Attn-only model. Bottom: local attention scores in the D-Attn model.

Yelp (user), L-Attn-only model: local attention

They carry some rare things that you can't find anywhere else. The staff is pretty damn cool too best in Arizona . I prefer ma-and-pa. They treat you the best and they value your business extreme . They are good people great atmosphere and music. I definitely believe that Lux has the best coffee I've ever had at this point. Screw all my previous reviews. This place has coffee down , they make damn good toast too .

Yelp (user), D-Attn model: local attention

They carry some rare things that you can't find anywhere else. The staff is pretty damn cool too best in Arizona. I prefer ma-and-pa. They treat you the best and they value your business extreme . They are good people great atmosphere and music. I definitely believe that Lux has the best coffee I've ever had at this point. Screw all my previous reviews. This place has coffee down, they make damn good toast too .

Table 5 shows words highlighted by the global attention scores from both the G-Attn-only model and the D-Attn model. In this example, we highlight *less important* words instead: words with lowest scores are red-colored, low scoring words are light-red, and high/medium score words are not colored. It seems that the global attention layer down-weighs neutral words and the later max pooling layer removes them from contributing to the final output. However, the G-Attn-only model does not do a perfect job: preference words like "rare", and "best" are grouped with their neighboring pronouns with very low scores. With the help of the local attention in the D-Attn model, the global attention in the D-Attn model synergizes with the local attention in the D-Attn by focusing on removing just the neutral words, while the local attention in the D-Attn emphasizes those preferences or attribute words.

Table 5: Highlighted words by global attention scores in a user's review. Top: attention scores in the G-Attn-only model. Bottom: global attention scores in the D-Attn model.

Yelp (user), G-Attn-only model: global attention

They carry some rare things that you can't find anywhere else . The staff is pretty damn cool too best in Arizona. I prefer ma- and -pa. They treat you the best and they value your business extreme. They are good people great atmosphere and music. I definitely believe that Lux has the best coffee I've ever had at this point. Screw all my previous reviews. This place has coffee down, they make damn good toast too.

Yelp (user), D-Attn model: global attention

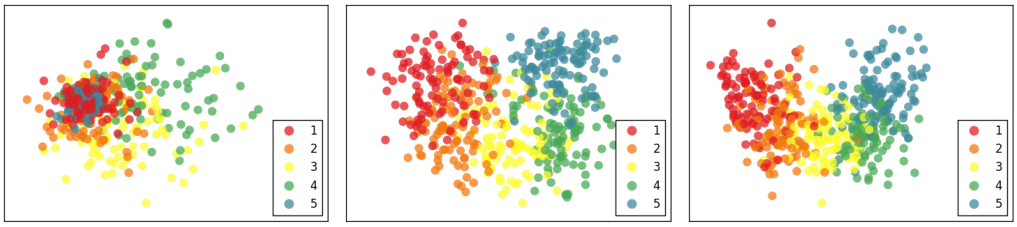
They carry some rare things that you can't find anywhere else . The staff is pretty damn cool too best in Arizona. I prefer ma-and-pa. They treat you the best and they value your business extreme. They are good people great atmosphere and music. I definitely believe that Lux has the best coffee I've ever had at this point. Screw all my previous reviews. This place has coffee down, they make damn good toast too.

Table 6 shows the same text but highlighted differently by the L-Attn model in the user's network and the item's network, respectively. More words in the review are highlighted by the user network to show the user's preferences.

5.3 Embedding Analysis

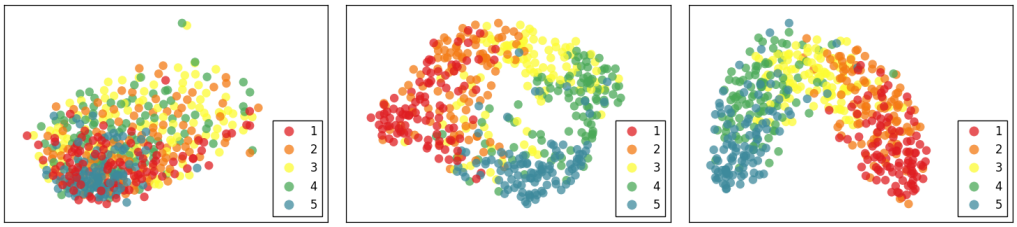
Our D-Attn model maps a user or an item to a latent representation before taking the dot product for rating estimation. We consider these representations as embeddings ([20, 21]) of each user/item. We show next that these pre-trained embeddings can be used for other tasks as well.

First, we examine whether these embeddings are meaningful and useful. If the embeddings describe users or items correctly, we



(a) HFT (K=50) (b) ConvMF+ (K=50) (c) D-Attn (K=50)

Figure 2: LDA projection of sampled item vectors of the Yelp data.



(a) HFT (K=50) (b) ConvMF+ (K=50) (c) D-Attn (K=50)

Figure 3: t-SNE embedding of sampled item vectors of the Yelp data.

Table 6: Attention words of the same review. Each network highlights different words.

Yelp (user), D-Attn: local attention
A disappointing meal and a very disappointing service . I won't be coming back anytime soon . If I was the manager I would demote the waiter and promote the busboy to a waiter as he was great tonight and he was the only reason I gave a 20% tip as it is unfair for him to suffer because of the waiters.
Yelp (item), D-Attn: local attention
A disappointing meal and a very disappointing service. I won't be coming back anytime soon. If I was the manager I would demote the waiter and promote the busboy to a waiter as he was great tonight and he was the only reason I gave a 20% tip as it is unfair for him to suffer because of the waiters.

can easily find similar users of a given user or most recommendable items to a given user. In Figure 3, we show embeddings of users and items that are trained with their ratings. Therefore, we can safely assume that items' embeddings are strongly related to their overall ratings. In other words, high-rated items would have more similar embeddings to each other than those of low-rated items. This assumption is more reasonable for an item embedding than a user embedding because ratings of a user are more dependent on what the user has purchased (or visited).

To examine the quality of item embeddings learned from HFT, ConvMF+ and our model, we sample 500 items (100 from each rating class) randomly from the Yelp dataset and classify the items (i.e. restaurants) to check whether the embeddings are good features. We apply the linear SVM for the multi-class classification of the embeddings (all with feature dimension of $K = 50$).

Table 7: Accuracy of multi-class classification.

	HFT	ConvMF+	D-Attn
Accuracy	0.482	0.580	0.602

Table 7 shows classification accuracies from three models. As we expected, both ConvMF+ and D-Attn have shown similar accuracies and D-Attn is slightly better than ConvMF+. HFT has the lowest accuracy, and it implies that the item representations of HFT are not strongly related to the average rating.

We then visualize these item embeddings in two-dimensional space to investigate subtle differences of the item embeddings. Figures 2 and 3 show the distributions of the two dimensional item embedding vectors. The top figures are obtained by Linear Discriminant Analysis (LDA) and the bottom figures are from t-distributed Stochastic Neighbor Embedding (t-SNE) with 30 perplexity [16]. Each point corresponds to each item in the dataset and the denoted number for each color is the average rating of each item.

Again, we see the item embeddings from HFT are not related to their average ratings. On the other hand, both ConvMF+ and D-Attn successfully cluster items together with the same average ratings: in the two dimensional space the five clusters are fairly distinguishable. This supports our assumption that similarly rated items are likely mapped to a closer embedding space. However, we still see the differences of the five clusters coming from ConvMF+ and our model: first, items of ConvMF+ with an average rating of 3 are more scattered over other classes than that of D-Attn (See Figure 3 (b) and (c)). This implies that D-Attn is more powerful in capturing differences between the mid-rated items and others. Second as the bottom figures show, we can see that the class (rating) distributions from D-Attn are gradually overlapped between closer

Table 8: List of recommended items

			HFT	ConvMF+	D-Attn
Query	Visited items	Rank	Recommendations [Categories]	Recommendations [Categories]	Recommendations [Categories]
user1	**The Duce** [Bars, Nightlife, Lounges, Restaurants]	1	**Yoshi's Asian Grill** [Asian Fusion, Restaurant]	**FnB** [Gastropubs, American]	**O'connor's Pub** [Pubs, Bars, Nightlife]
		2	**Nathans Famous Hotdogs** [Hot Dogs, Restaurants]	**Petite Maison** [French, Restaurants]	**Rosie McCaffrey's** [Pubs, Bars, Nightlife, Restaurants]
		3	**Wendy's** [Fast Food, Restaurants]	**Lox Stock & Bagel** [Bagels, Breakfast & Brunch, Restaurants]	**The Vig** [Pubs, Bars, Nightlife, Restaurants]
		4	**The Coffee Bean & Tea Leaf** [Food, Coffee & Tea]	**True Food Kitchen** [American, Restaurants]	**Arcadia Tavern** [Pubs, Bars, Nightlife, Sports Bars]
user2	**Matador Restaurant** [Mexican, Greek, Restaurants]	1	**The Saguaro** [American, Mexican]	**Rocket Burger & Subs** [Burger, Hot Dogs, Sandwiches]	**Sofia's Mexican** [Mexican, Restaurants]
		2	**The Grapevine** [American, Karaoke]	**Roka Akor** [Steakhouses, Sushi Bars, Japanese]	**Tacos Jalisco** [Mexican, Restaurants]
		3	**Citizen Public House** [Gastropubs, American]	**The Fry Bread House** [American, Restaurants]	**Carolina's Mexican** [Mexican, Restaurants]
		4	**AZ 88** [Bars, American, Lounges]	**Five Guys** [Burgers, Restaurants]	**El Taco Tote** [Mexican, Restaurants]

ratings (e.g., rating 1 and 2, not 1 and 5) unlike the distributions from ConvMF+.

5.4 Recommendation

We confirm that an embedding of a given item learned from our model is correlated with the average rating of the item. Now, we investigate the possibility of using the user/item embeddings for practical recommendations. We use the Yelp dataset which has detailed information about users and items (restaurants).

First, we pick a random user who has a single review because we are going to recommend potential items to someone who does not have enough review history. The selected user is considered a query user. To recommend potential items, we first find the *nearest neighbors* to the query user in the user embedding space. Our hypothesis is that the nearest neighbors apparently have similar favorites or tastes with the query user, hence, items visited by the nearest neighbors would also be highly likely preferred by the query user.

In Table 8, we sample two users who visited only one restaurant and gave a positive rating (higher ratings than 3). We carry out the following procedure to come up with a recommended list of restaurants: first, given a query user, we find the K nearest neighbors in the user embedding space and the items that were already visited by these neighbors as candidates. Second, we compute potential ratings by the query user on these candidates. Finally, we sort the items by the rating estimations to select the most preferred ones. In Table 8, we list four restaurants from a set of candidate restaurants visited by $K = 5$ nearest neighbors.

Surprisingly there are big differences among the recommendation lists returned by HFT, ConvMF+ and D-Attn. The results show that the recommended restaurants by D-Attn and the *already visited* restaurants by the query users have common categories unlike other models. For example, D-Attn recommends *Mexican* restaurants to user2 who has already visited *Matador Restaurant* which serves Mexican foods. Moreover, we manually check that those four recommendations have common features (food menu), such as price (one $ in Yelp), or atmosphere (casual ambience and attire). Hence, these results suggest that the embeddings of user and item from the D-Attn model could be useful for the nearest-neighbors recommendation.

6 CONCLUSIONS

We have presented an interpretable, dual attention-based CNN model, D-Attn, that combines review text and ratings for product rating prediction. It learns the latent representation of users and items from the aggregated reviews, and enables the interaction between user and item models in a way similar to matrix factorization. By leveraging the best of both collaborative filtering and topic-based approaches, our model is inherently more robust to noise and inconsistency in the review and rating data by applying both local and global attention layers and combining these two in one network training. Our model is validated by experiments on both the Yelp and Amazon datasets. Our model outperforms HFT and the recently proposed ConvMF+ in terms of prediction errors. Furthermore, we show that these latent representations of users and items are meaningful and effective to provide practical applications for recommendation.

We are convinced that this work offers a new avenue to apply representation learning in the context of recommendation systems. One future direction is to use sequence learning, e.g., combine LSTM with attention network, to handle long-range dependency in the review text.

7 ACKNOWLEDGMENTS

This work is supported in part by NSF award number IIS-1254206 and Facebook Research Award. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government.

REFERENCES

[1] Amjad Almahairi, Kyle Kastner, Kyunghyun Cho, and Aaron Courville. 2015. Learning distributed representations from reviews for collaborative filtering. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 147–154.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. *International conference on learning representations* (2015).

[3] Yang Bao, Hui Fang, and Jie Zhang. 2014. TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation.. In *AAAI*. 2–8.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[5] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very Deep Convolutional Networks for Natural Language Processing. *arXiv preprint arXiv:1606.01781* (2016).

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 191–198.

[7] Cícero Nogueira dos Santos and Maira Gatti. 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts.. In *COLING*. 69–78.

[8] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *Proceedings of the 10th ACM Conference on Recommender systems*. ACM, 223–240.

[9] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615* (2015).

[10] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

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

[12] Jiwei Li, Minh-Thang Luong, Dan Jurafsky, and Eudard Hovy. 2015. When are tree structures necessary for deep learning of representations? *arXiv preprint arXiv:1503.00185* (2015).

[13] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

[14] Guang Ling, Michael R Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 105–112.

[15] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*.

[16] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[17] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 165–172.

[18] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.

[19] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[21] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[22] Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 959–962.

[23] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 101–110.

[24] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, Vol. 1631. 1642.

[25] Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proc. ACL*.

[26] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[27] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *Proceedings of the 32nd International Conference on Machine Learning* (2015).

[28] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[29] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*. 649–657.

[30] 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 *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 83–92.

[31] Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* (2015).