Hierarchical Attention based Neural Network for Explainable Recommendation

Dawei Cong
Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology
Harbin, China
dwcong@ir.hit.edu.cn

Yanyan Zhao*
Department of Media Technology and Art, Harbin Institute of Technology
Harbin, China
yyzhao@ir.hit.edu.cn

Bing Qin
Peng Cheng Laboratory
Shenzhen, China
bqin@ir.hit.edu.cn

Yu Han
Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology
Harbin, China
yhan@ir.hit.edu.cn

Murray Zhang
Alden Liu
Nat Chen
Tencent, AMS, WXAD
Shenzhen, China
{murrayzhang,aldenliu,natchen}@tencent.com

ABSTRACT

In recent years, recommendation systems have attracted more and more attention due to the rapid development of e-commerce. Reviews information can offer help in modeling user's preference and item's performance. Some existing methods utilize reviews for the recommendation. However, few of those models consider the importance of reviews and words in corpus together. Therefore, we propose an approach for rating prediction using a hierarchical attention-based network named HANN, which can distinguish the importance of reviews at both word level and review level for explanations automatically. Experiments on four real-life datasets from Amazon demonstrate that our model achieves an improvement in prediction compared to several state-of-the-art approaches. The hierarchical attention weights in sampled test data verify the effect on selecting informative words and reviews.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommendation Systems, Neural Networks, Explainable Recommendation

ACM Reference Format:
Dawei Cong, Yanyan Zhao, Bing Qin, Yu Han, Murray Zhang, Alden Liu, and Nat Chen. 2019. Hierarchical Attention based Neural Network for Explainable Recommendation. In *International Conference on Multimedia Retrieval (ICMR '19), June 10–13, 2019, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3323873.3326592

*This author is the corresponding author.

ICMR '19, June 10–13, 2019, Ottawa, ON, Canada

ACM ISBN 978-1-4503-6765-3/19/06…$15.00
https://doi.org/10.1145/3323873.3326592

1 INTRODUCTION

With the continuous expansion of e-commerce, the recommendation systems have become more and more important in helping customers make decisions from the rapidly increased products. Recently, explainable recommendation technology has attracted many researchers' attention[4][11], because it can give the evidence that can explain why recommending an item to a user. This technology with explainable characteristic can be more humanized and personalized compared to the traditional collaborative filtering technologies which are based on the behaviors of the users and the items.

Reviews which have rich and useful user-generated content are always used as the evidence for the recommendation explanations. For example, SULM[2] obtains information from reviews by external sentiment analysis tools, EFM[32] extracts explicit product features (i.e. aspects) and user opinions by phrase-level sentiment analysis on user reviews for recommendation. TriRank[14] enriches the user-item binary relation to a user- item-aspect ternary relation (i.e., the specific properties of items) with the help of aspect extraction tool used in textual reviews. sCVR[24] employs a sentiment analysis method to classify user reviews into positive and negative categories. The probability of a sentiment label is set as a prior value in their method. Some researchers[11] builds an initial network based on Microsoft Concept Graph and improves the model accuracy by optimizing key variables in the hierarchy. DeepCoNN[37] attempts to gain features automatically and treats every review equally. [4] uses a review-level attention mechanism to explore the usefulness of reviews. Although these methods are effective, they still have some limitations.

Firstly, most of the previous work considered that all the reviews of a user/item have the same importance for predicting the final rating. Actually, different reviews have different importance. For instance, some reviews are very informative and can provide rich evidence for the recommendation systems. Thus these reviews are more helpful and correspondingly we should pay more attention to them. Secondly, in the same review, different words also have different importance for predicting the final rating. For instance, the polarity word and the aspect may be more important than the other words in the review. But that not means words selected by external tools or manual features can summarize the semantics of reviews completely and accurately. That is to say, a better way is that these comment relative words should get more attention.

To overcome these two limitations, we design a hierarchical attention framework to learn the interaction between users and items from reviews to infer the rating and construct an explainable recommendation system. There are two layers in the hierarchical attention framework. The first layer is word-level, which works with intra-review attention. This kind of attention is used to obtain the different importance of the words in the same review. The second layer is review-level, which works with inter-review attention. This kind of attention is used to capture the different importance of reviews for one product. We use the user-item interaction to distinguish the importance of reviews at both word level and review level for explanations automatically.

To combine these two levels of information, in this paper we propose a hierarchical attention based neural network named HANN for explaining rating prediction. Thus, this hierarchical structure with different review weights and different word weights can naturally show the explanations for the recommendation system. Our experiments are conducted on four real-life datasets from Amazon. The experimental results for these datasets show that our proposed HANN model is consistently better than all the baseline methods on all benchmark datasets. This demonstrates the two types of review attentions are effective and useful, and further our hierarchical attention based neural network framework is reasonable and well-designed. It is worth noting that HANN outperforms NARRE, which is the recent state-of-the-art review-based methods for explainable recommendation without external tools.

The contributions of the paper are as follows.

- We propose two kinds of review attentions, namely, intra-review attention and inter-review attention. The first one can reflect the word difference in a review, and the latter one can explore the importance of different reviews towards a user/item.
- We present a framework of hierarchical neural network named HANN to integrate the two kinds of review attention. HANN not only considers the usefulness of reviews at review level but also at word level. The well-designed hierarchical attention mechanism helps the model capture user profiles and item profiles, making them more explainable and reasonable, and ultimately leads to improvements in rating prediction.

This paper is organized as follows: Section 2 introduces the related work on explainable recommendation systems; Section 3 details the two types of review attention and the hierarchical neural network framework HANN for explainable recommendation; Section 4 describes the experimental setup and results; Section 5 makes additional experiment to evaluate how each part of our component contributes to our full model and then gives an explanation analysis of our model HANN; and finally, the conclusion is in Section 6.

2 RELATED WORK

2.1 Deep Learning-based Recommendation

Widely used in both research and industry communities, recommendation has received lots of researchers' attention. Many methods including content-based[19], collaborative filtering-based[8] and hybrid methods[9] have been proposed to improve recommendation performance. Recently, deep neural networks have been successfully applied to a large variety of tasks, such as speech recognition, image captioning[36] and natural language processing[12], and have achieved good results. Many proposed recommendation models have combined neural network with traditional methods to further improve accuracy. Generally speaking, there are two types of deep matching models for recommendation. One is based on representation learning. [27] worked matrix factorization as a neural network for learning the user and item embeddings. Other is using neural networks to learn the matching function for user-item interaction. For example, [17] presented a Neural Collaborative Filtering (NCF) framework to learn the nonlinear interactions between users and items. Moreover, [16] proposed to use outer-product based NCF. Later, Neural Factorization Machines(NFM)[15] enhanced FM by modelling higher-order and non-linear feature interactions.

2.2 Review-based Recommendation

In recent years, researchers have discovered a new research field, explainable recommendation, which usually extracted recommendation reasons from reviews. More specifically, early methods such as SULM[2], EFM[32], TriRank[14] and sCVR[24], mainly draw support from external tools or manual features because sentiment analysis may help refer user's characteristics[35][33]. [11] built an initial network based on Microsoft Concept Graph and improved the model accuracy by optimizing key variables in the hierarchy. Although these work has achieved considerable success in improving explainability, there is a limitation that their results may rely on the accuracy of their external tools or manual features. In DeepCoNN[37], convolutional neural networks were leveraged to process textual reviews and extract features for rating prediction by two parallel parts coupled in the last layers. Different from DeepCoNN treating every reviews equally, NARRE[4] paid attention to the usefulness of reviews by its attention mechanism, but it can not extract useful words. Compared with these methods, we distinguish the usefulness of both reviews and words, which is more detailed and explainable without external tools or manual features.

2.3 Neural Attention Mechanism

Loosely based on the visual attention mechanism found in humans, the attention mechanism in neural networks has been shown effective in various machine learning tasks such as image captioning [5][34], neural machine translation[1] and document classification [30]. A big advantage of attention is that it provides neural networks with guidance, parts with higher weights contain more informative features and should be noticed more. It has also been applied in recommendation for seeking the important and useful parts, such as specific features, words, sentences in textual reviews. He et al.[6] introduced an attention mechanism in CF which consists of both component-level and item-level attention module for recommendation, which did not care about explanation. Seo et al.[25] combined local and global attention to enable an interpretable and better-learned representation of users and items, which can select some useful words. [4] used a review-level attention mechanism to explore the usefulness of reviews. In this paper, the attention mechanism is designed to work with hierarchical neural network. It equips the network with the ability to focus on informative words and reviews for the prediction and explanations.

3 APPROACH

In this section, we introduce our proposed Hierarchical Attention based Neural Network for explainable recommendation (HANN), which aims to capture the interaction between users and items from reviews to infer the rating, as well as give an explanation at both word level and review level. First, we will present the general architecture of HANN. Follow which, we describe in detail user-item interaction representation and our text processing module for learning the user/item representations. Next, we will show the hierarchical attention used in our model, which is the main concern in this paper. Then, we will introduce the prediction layer, which contains information of user-item pair and their own profile to predict. Lastly, we will go through the optimization details of HANN.

3.1 Overview of HANN

Considering a corpus of ratings R and reviews D, for a set of items I and a set of users U, and $r_{u,i}$ is a numerical rating denoting user u's overall satisfaction towards item i, and $d_{u,i}$ is the corresponding textual review. The target of our model is to estimate the rating $r_{u,i}$ for unseen user-item pair with no interaction, as well as to select both useful and representative words and reviews. It should be noted that $d_{u,i}$ is not included in the input data when predicting $r_{u,i}$ because considering the actual situation, $d_{u,i}$ is not existing at the inference stage. $r_{u,i}$ usually depends on u's preference, i's performance and whether i is suitable for u, which could be reflected in their accompanying reviews. Based on this, we propose a model to learn both interaction of user-item pair and their own attributes.

The architecture of the proposed model is shown in Figure 1. The model consists of three modules, the center module for capturing information of user-item pair ($Net_{u,i}$), two parallel neural networks, one for user modeling (Net_u), and the other for item modeling (Net_i). Since Net_u and Net_i only differ in their inputs and the processes applied for two modules are the same, we focus on illustrating the process for Net_u in details. In the following, we will introduce user-item interaction modeling, hierarchical text processor, attention mechanism working in the network for extracting useful words and reviews and how to combine user-item interaction and their profiles to predict ratings.

3.2 User-item Interaction

To facilitate the information seeking process for user-item relationships, early recommendation researches mapped users and items to latent factor spaces[27][17]. There are two main types of methods to measure the degree of matching between user and item, one is calculating user and item representation, then conduct matching; the other is constructing basic low-level matching signals and aggregating matching patterns. Here we choose the latter.

We utilize embedding matrix for user features and item features, $V_U \in \mathbb{R}^{M\times K}$ and $V_I \in \mathbb{R}^{N\times K}$, respectively; K, M, and N denote the size of embedding, number of users, and number of item, respectively. And let v_u and v_i be the embeddings of u and i, respectively. For each user-item pair, we employ their element-wise product to model interactions of the user and the item.

$$v_{u,i} = v_u \odot v_i \tag{1}$$

where $v_{u,i}$ is a vector in the same size of user/item embedding.

3.3 Hierarchical GRU Text Processor

In addition to the vector representation, texts also provides information for user/item features. In recent years, many text processing methods based on deep learning technology have been applied and have achieved an ideal result. A word embedding layer maps each word in the review into a d dimensional vector and then transforms the given review into an embedded matrix. The embedding can be any pre-trained embedding like those trained on GoogleNews corpus using word2vec[1][21], or on Wikipedia using GloVe[2][23]. The embedded matrix of reviews will be input to a hierarchical GRU network for obtaining information about user profile from reviews at word-level and review-level. GRU[7] is a related variant of RNN, which is able to handle a variable-length sequence input. Compared with CNN, GRU uses activation which is dependent on that of the previous time at each time and can learn semantic information better. As for our hierarchical model, two layers of GRU is employed to extract information from reviews. The first GRU which is called intra-review GRU focuses on acquiring semantics in each review, while the second GRU which is called inter-review GRU makes a summary and learns user's preference and representation. First, we put the embedded matrix of each review written by the user into the intra-review GRU, which outputs the hidden state vectors $h_1, h_2, ..., h_n$ for the reviews, where n is the length of the review. h_j matches ith word in the review and contains its context meaning, a general idea is that their average h stands for the meaning of the whole review.

$$h = \frac{1}{n} \sum_{j=1,\dots,n} h_j \tag{2}$$

We then feed average hidden vector h of each user's review into the inter-review GRU and aim at learning the user's preference and modeling its profile. The output of the second GRU is denoted as $s_1, s_2, ..., s_m$, where m is the number of the user's reviews. Similarly, we aggregate these vectors to get the representation of the user.

$$s = \frac{1}{m} \sum_{j=1,\dots,m} s_j \tag{3}$$

Now we process all the user's reviews and gain the user's feature vector. It should be noted that computing methods of Eq. 2, Eq. 3 assume each word and each review with equal effects on user's expression. However, this assumption is obviously not in line with common sense. To alleviate this problem, we introduce the attention

[1]https://code.google.com/archive/p/word2vec/
[2]https://nlp.stanford.edu/projects/glove/

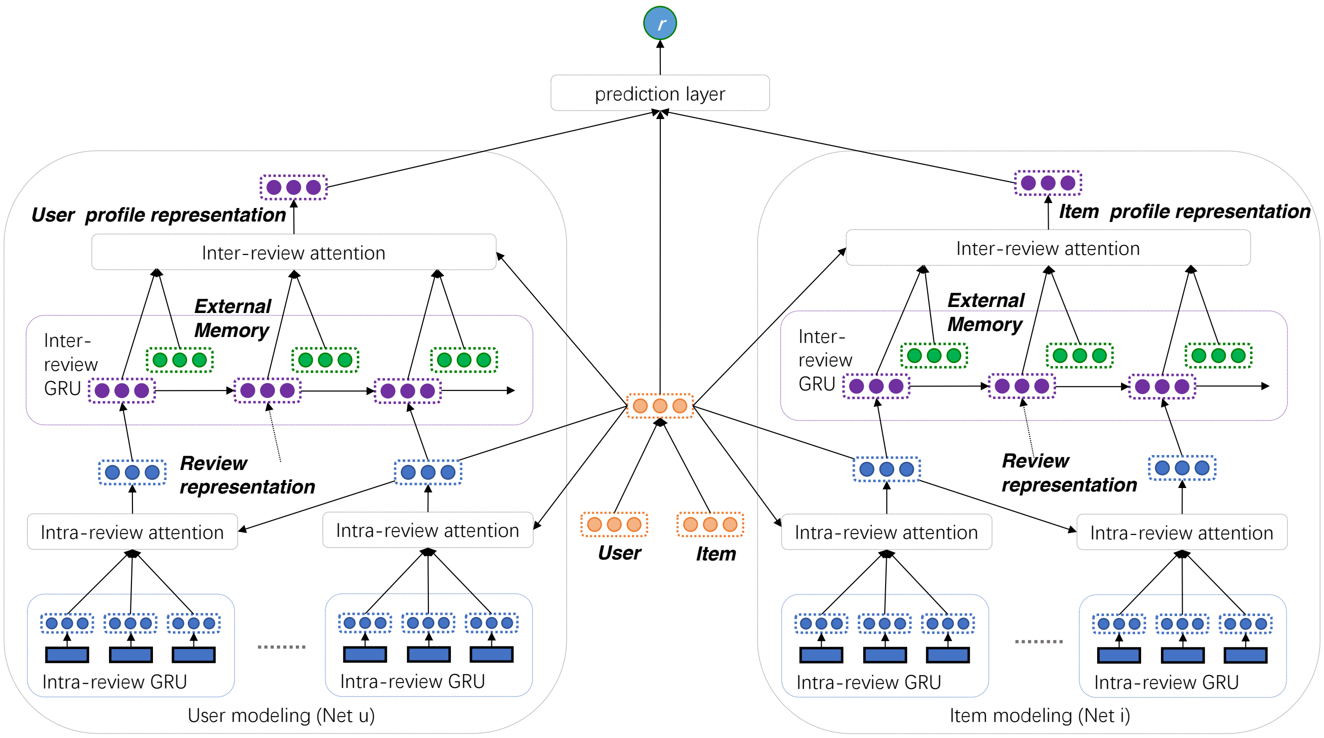

Figure 1: Overview of HANN. Our hierarchical attention based neural network uses intra-review attention and inter-review attention to dynamically assign weights to words and reviews respectively.

mechanism into our model, which can distinguish the importance of different words and reviews.

3.4 Attention Mechanism for Explaining Rating Prediction

The attention mechanism is a very effective method of leveraging context-aware features over variable-length sequences, which has achieved good results in many NLP tasks, such as information retrieval[29], recommendation[28] and machine translation[10]. In essence, it equips neural networks with the ability to focus on selective parts of the input. Our primary objective is focusing on words and reviews with more importance to infer user's overall satisfaction, while attention mechanism is suitable for this task. As described in [26], an attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. We will introduce the attention mechanism used in our hierarchical model in this form below. To deal with different situations, we employ corresponding methods. Specifically speaking, we should compare different words' importance in intra-review and reviews' in inter-review.

3.4.1 Attention for intra-review. The model should learn which words are more informative in a review. At this level, informative words stand for containing features of user and item. We map user-item interaction vector $v_{u,i}$ to query and intra-review GRU hidden vector h_j to key and value in the intra-review attention function. First, we compute weighting scores for each word in the review as follows:

$$\alpha_j^* = W_a^T ReLU(W_h h_j + W_u v_{u,i} + b_1) + b_2 \tag{4}$$

where W_a, W_h, W_u, b_1, b_2 are model parameters, and ReLU[22] is a nonliner activation function. Then a softmax function is used to normalizing the above attention scores:

$$\alpha_j = \frac{exp(\alpha_j^*)}{\sum exp(\alpha_j^*)} \tag{5}$$

Now we get the weight of each word in the review which reflects the importance of each word for the user-item pair, we will replace h in the previous section with the weighted sum as follows:

$$h = \sum_{j=1,\ldots n} \alpha_j h_j \tag{6}$$

3.4.2 Attention for inter-review. Now we consider how to select reviews that are representative to the user's feature. Each review corresponds to an item, and both user's reviews and the item purchased by the user are helpful to capture the user's preference. To better represent information of the item, we use an additional embedding matrix for item features, $P_I \in \mathbb{R}^{N\times K}$ which is called external memory, K, and N denote the size of embedding and number of items, respectively. And let p_j be the external memory vector of item j. We map user-item interaction vector $v_{u,i}$ to query and the concatenation of inter-review GRU hidden vector s_j and external memory vector p_j to key and value in the inter-review attention function. First, we compute weighting scores for each review as follows:

$$\beta_j^* = W_b^T ReLU(W_s(s_j \oplus p_j) + W_v v_{u,i} + b_3) + b_4 \tag{7}$$

where W_b, W_s, W_v, b_3, b_4 are model parameters, and ReLU is a nonliner activation function. Then we also use the softmax function to gain attention scores of each review:

$$\beta_j = \frac{exp(\beta_j^*)}{\sum exp(\beta_j^*)} \tag{8}$$

And replace s in the previous section with the weighted sum as follows:

$$s = \sum_{j=1,\dots n} \beta_j(s_j \oplus p_j) \tag{9}$$

3.5 Prediction Layer

As mentioned above, we gain user-item interaction vector $v_{u,i}$ and their own representation vector s_u and s_i (the output of Net_u and Net_i). All of those representation vectors are concatenated and run through additional fully connected layers to transform dimension for the next step. The calculation process is as follows:

$$c = W_c(s_u \oplus s_i \oplus v_{u,i}) + b_c \tag{10}$$

where W_c, b_c are weights, bias of the fully connected layers, respectively. Then the output vector is fed into the prediction layer to get a real-valued rating $r_{\hat{u},i}$ as follows:

$$r_{\hat{u},i} = W_1^T c + b_u + b_i + \mu \tag{11}$$

where $W_1 \in \mathbb{R}^n$ denotes weights of the prediction layer, b_u, b_i, μ are the user, item, and global bias, respectively.

3.6 Learning

Now we describe the learning process that trains the model in an end-to-end fashion. Since our task is rating prediction, which actually is a regression problem. For regression, an objective function, the squared loss is commonly used. Besides, many existing works have found that machine learning models tend to suffer from overfitting. There are many methods widely adopted in existing models in order to improve the generalization performance at present. Regularization is the process of adding information in order to solve an ill-posed problem or to prevent overfitting[3] in mathematics, statistics, and computer science, particularly in machine learning and inverse problems. And it applies to objective functions in optimization problems. Our final objective function to be minimized is:

$$L_r = \sum_{u,i}(r_{\hat{u},i} - r_{u,i})^2 + \gamma \sum_{\theta \in \Theta} ||\theta||_2^2 \tag{12}$$

where Θ is the set of parameters to be reguarized. The meaning of the equation is that the first term aims to minimize the distance between the real and the predicted ratings, while the second term regularizes the parameters to avoid overfitting.

Table 1: Statistical details of the datasets

Dataset	Musical	Toys	Clothing	Home
users	1429	19412	39387	66519
items	900	11924	23033	28237
ratings & reviews	10261	167597	278677	551682

4 EXPERIMENTS

In this section, We have performed extensive experiments on Amazon datasets to demonstrate the effectiveness of HANN compared to other state-of-the-art recommendation systems. We first describe the datasets used in our experiments in Section 4.1. The evaluation method and baselines algorithms selected for comparisons are explained in Section 4.2. Implementation details are given in Section 4.3. Empirical results are discussed in sections 4.4.

4.1 Dataset

In our experiments, We utilize publicly available datasets to evaluate our model, which are described as follows:

- **Amazon Product Data**[3]: Amazon is a well-known Ecommerce platform. Users are able to write reviews for the products they have purchased. This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. It has been investigated by many researchers[13]. As far as we know, this is the largest publicly accessible rating dataset with text reviews.

In this paper, we focus on training interpretable models rather than dealing with the cold start issues. Therefore, we start our experiments from the 5-core subset, such that each of the remaining users and items has at least 5 reviews. The raw data is divided into 24 subsets. In order to cover both different domains and different scales, we select four categories in our experiments, that is **Musical Instruments**, **Toys and Games**, **Clothing Shoes and Jewelry**, **Home and Kitchen**. Among them, Home and Kitchen is the largest dataset and it contains more than 0.5 million reviews, while Musical Instruments is the smallest one and only contains about 10 thousand reviews. The key characteristics of these datasets are summarized in Table 1.

4.2 Evaluation Method and Baselines

Amazon product data is rated as integers in the range [1,5], therefore we adopt the well-known Root Mean Square Error (RMSE) which is widely used for rating prediction in recommendation systems to evaluate the performance of the algorithms. The lower the RMSE score, the better the performance. Based on a predicted rating $r_{\hat{u},i}$ and a ground-truth rating $r_{u,i}$ from the user u for the item i, the RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{u,i}(r_{\hat{u},i} - r_{u,i})^2} \tag{13}$$

where n indicates the number of ratings between user-item pairs. To validate the effectiveness of HANN, we have selected three state-of-the-art algorithms as comparisons to our proposed model for

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

Table 2: Comparison of the Approaches

	NeuMF	DeepCoNN	NARRE	HANN
Ratings	✓	✓	✓	✓
Textual Reviews		✓	✓	✓
Review-level Usefulness			✓	✓
Word-level Usefulness				✓

evaluations: NeuMF, DeepCoNN and NARRE. The first method only uses ratings, while the latter two incorporate the valuable information in user-generated textual reviews into rating prediction. Compared to DeepCoNN, NARRE learns the usefulness of each review but is limited to the review level. Our proposed method HANN not only considers the usefulness of reviews at review level but also at word level. The key characteristics of the comparative approaches are listed in Table 2.

- **NeuMF**[17]: Neural Matrix Factorization is the state-of-the-art model for interaction-only CF. This model combines the linearity of GMF and non-linearity of MLPs for modeling user-item latent structures. Due to the Amazon Data Product datasets are based on explicit feedback, we changed the original 0-1 classification task to a rating prediction task. We implemented it based on the authors’ public code[4].
- **DeepCoNN**[37]: Deep Co-Operative Neural Networks is a method that uses deep learning techniques to jointly model users and items from textual reviews. This approach has significant improvements over other strong topic modeling based methods. We implemented it based on the authors’ public code[5].
- **NARRE**[4]: Neural Attentional Regression model with Review-level Explanations is a state-of-the-art interpretable recommendation method that has proven to be superior to many promising algorithms including NMF, SVD++ and HFT on Amazon datasets. We implemented it based on the authors’ public code[6].

4.3 Implementation Details

Firstly, following previous work[4], we randomly divide each of these four datasets into training, validation, and test sets using a ratio of 80:10:10. The validation set is used for tuning hyper-parameters and the final performance comparison is performed on the test set.

Next, we use NLTK to tokenize the reviews. Just like the method in NARRE, when we build user and item profile, because of the length and the number of reviews have a long tail effect, we only keep the length and the number of reviews covering p percent users and items respectively, the p is set to 90 for Musical Instruments, Toys and Games, Clothing Shoes and Jewelry and Home and Kitchen. We would like to emphasize that, when building user and item profile using their respective reviews, all reviews belonging to interactions from the test and development sets are not included.

We implement our model in Tensorflow. We use a variant of the Adam[18] optimizer called AdamW[7][20] which includes "correct" L2 weight decay with initial learning rate of 0.001, $\beta 1$ = 0.9, $\beta 2$ = 0.999, $\epsilon = 10^{-6}$, learning rate warmup over the first 10% steps, and linear decay of the learning rate over the rest steps. The L2 weight decay is searched in [0.0005, 0.001, 0.002, 0.004], the batch size is searched in [16, 32], and the dropout rate is fixed at 0.5. For our proposed model, the latent factors number K is searched in [16, 32, 64, 128, 256], and we will discuss its impact on model performance in the following sections.

The optimization method and parameter initialization of the baseline algorithm is as described in the corresponding paper. For NeuMF, we set the number of latent factors K=64 and use MLP-2 which indicates the MLP method with two hidden layers. For DeepCoNN and NARRE’s CNN Text Processors, we reuse most of the hyper-parameter settings reported by the DeepCoNN authors, because changing them does not provide any perceivable improvement, the number of filters, m, in the convolutional layer is 100, the filter size t is 3. In addition, we use a pre-trained 300-d word embeddings which are trained on more than 100 billion words from Google News[21] for all methods.

4.4 Results

Table 3 shows the rating prediction results of our model HANN and baseline methods on four datasets. Based on the results, the following conclusions can be drawn:

Firstly, reviews-based methods (DeepCoNN and NARRE) always perform better than interaction-only models (NeuMF) which only consider the user and item embedding as the input. These models incorporate valuable information from user-generated text reviews into the user modeling and recommendation process. These reviews are usually in the form of textual comments, explaining why they like or dislike an item based on their experience. The system can capture the multi-faceted nature of a user’s opinions from reviews, thereby building a fine-grained preference model for the user, which however cannot be obtained from overall ratings.

Secondly, regarding the relative ranking of the review-based models, our empirical evaluation reiterates the claim of[4], showing that NARRE is always better than DeepCoNN. Although review information is helpful for recommendations, the performance may vary depending on how the review information is utilized. Compared to DeepCoNN, NARRE learns the usefulness of each review which can lead to a better performance according to the results.

Finally, as shown in Table 3, we observe that our proposed HANN model is consistently better than all the baseline methods on all benchmark datasets. This clarifies the effectiveness of the model we propose. HANN outperforms NARRE, which is the recent state-of-the-art review-based methods for recommendation. The relative improvement against NARRE are 1% (Musical Instruments), 0.4% (Toys and Games), 0.3% (Clothing Shoes and Jewelry) and 0.4% (Home and Kitchen) respectively. The relative improvement against DeepCoNN are 1.2% (Musical Instruments), 1.1% (Toys and Games),

[4]https://github.com/hexiangnan/neural_collaborative_filtering
[5]https://github.com/Quincy1994/DeepCoNN
[6]https://github.com/chenchongthu/NARRE
[7]https://github.com/loshchil/AdamW-and-SGDW

Table 3: Comparison with state-of-the-art baseline methods in terms of the Root Mean Squared Error (The best result for each dataset is indicated in bold). The last column of the table represents the average RMSE of each method.

	Musical	Toys	Clothing	Home	Mean
NeuMF	0.9407	0.9037	1.0603	1.0565	0.9903
DeepCoNN	0.9385	0.8982	1.0707	1.0549	0.9906
NARRE	0.9363	0.8919	1.0530	1.0407	0.9805
HANN	**0.9268**	**0.8882**	**1.0497**	**1.0364**	**0.9753**

2% (Clothing Shoes and Jewelry) and 1.8% (Home and Kitchen) respectively. Notably, the average percentage improvement of HANN over DeepCoNN is 1.5% and the average performance gain over NARRE is a modest 0.5%. Compared to NARRE, Our proposed method HANN not only considers the usefulness of reviews at review level but also at word level. The well-designed hierarchical attention mechanism helps the model capture user profiles and item profiles, making them more explainable and reasonable, and ultimately leads to improvements in rating prediction.

5 HYPERPARAMETER & ANALYSIS

In section 4, improvements among all four datasets have been witnessed. However, we would like to know the reason for the improvement. In this section, we first explore the influence of the latent size K. Then, we provide meaningful insights on the effect of hierarchical attention mechanism, external memory and symmetric architecture respectively. We make additional experiments to evaluate how each part of our component contributes to our full model. Finally, we give an explanation analysis of our model.

5.1 Influence of the Latent Size K

We study the influence of different latent size K to the performance of our model in this subsection, and due to space constraints, we only show the results of the Musical Instruments dataset. We plot the results in Figure 2. We observe the performance changes by tuning the latent size K in the range of [16, 32, 64, 128, 256]. We see that the performances gradually increase with the decrease of latent size K, and they tend to be stable when the latent size is relatively small (i.e., $K = 64$), while larger K does not help to further improve the results. And this is why we adopt 64 as the default latent size in the previous experiments. This observation actually similar to many previous studies[31], probably because: in our dataset, a small number of parameters are enough for capturing the feature of user profile and item profile, using too many latent factors for model learning will increase the complexity of model and may even result in over-fitting in the training set, which may reduce the generalization performance on the test set.

5.2 Effect of Hierarchical Attention Mechanism

In our model, the hierarchical attention mechanism is a key component. We investigate how the hierarchical attention mechanism influences our model's performance in this section, and due to the space limitation, we only show the results on the Toys and Games dataset. For the purpose of verifying the effect and rationality of our hierarchical attention mechanism, two experiments that do away with word-level attention and review-level attention respectively are designed. The results without word-level attention or review-level attention are shown in Table 4. The table conveys the message that if we only adopt only one of them rather than provided the whole attention mechanism, we will obtain the worse results, which demonstrates the importance and effectiveness of the hierarchical attention mechanism. We attribute it to the reason that both word-level and review-level information works and contributes to rating prediction.

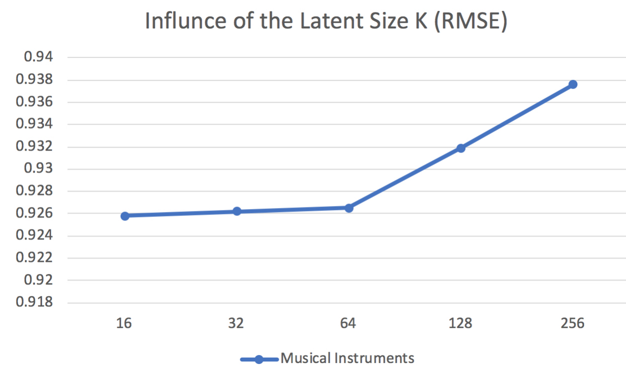

Figure 2: Influence of the Latent Size K. The performances gradually increase with the decrease of latent size K, and they tend to be stable when the latent size is relatively small.

Table 4: Effect of hierarchical attention mechanism. The performances of the model drop without word-level attention or review-level attention.

our model	only word-level	only review-level
0.8882	0.8893	0.8890

5.3 Effect of External Memory

In this section, we investigate how external memory influences our model's performance. Same as the previous section, we only report the results on the Toys and Games dataset.

In our model, external memory is a key component. In addition to user's reviews, we also use the representation of items purchased by the user which is called external memory for modeling user profile, and the same for item profile. We want to know whether utilizing it in the attention mechanism is helpful. So we conduct the experiments on two variant models, one is removing external memory from the key of inter-review attention(remove p_j from Eq. 7), the other is remove external memory from the value of inter-review attention(remove p_j from Eq. 9). Table 5 gives the result of the comparison experiment. From the result, we can observe that if we remove external memory from either key or value of inter-review attention, the performance of the rating prediction will be

Table 5: Effect of external memory. The performances of the model drop when the external memory is only in key or value.

our model	only in key	only in value
0.8882	0.8893	0.8888

Table 6: Effect of symmetric architecture. The performances of the model drop without user modeling or item modeling.

our model	only user modeling	only item modeling
0.8882	0.8899	0.8895

slightly dropped, which demonstrates that incorporating external memory both in key and value can help leverage the purchased history information for modeling user and item profile.

5.4 Effect of Symmetric Architecture

In this section, we investigate how symmetric architecture influences our model's performance. Same as the previous section, we only report the results on the Toys and Games dataset.

Our HANN model contains two parallel neural networks, one for user modeling (Net_u), and the other for item modeling (Net_i), we want to know whether the symmetric architecture is effective. So we designed two new models to experiment, one containing only the user model and the other containing only the item model. Results of these two models are shown in Table 6. The results demonstrate the effectiveness of the symmetric architecture to the final performance. The symmetric architecture captures both the user and item profile. Without one of them, the rating prediction performance will be harmed.

5.5 Explanation Analysis of HANN

To further show the advantages of our model, we randomly sample and visualize some examples from test data. The examples in Figure 3 are from one user. We highlight words and reviews with high word-level and review-level attention scores. Darker pink corresponds to reviews with higher scores and darker green corresponds to words with higher scores. As we can see from examples in Figure 3, the model extracts useful words in the review and distinguishes the influence of different reviews towards user-item rating prediction. It is not difficult to find that the user pays more attention to price and quality. The first reviews contain most description about price and the item's performance, which obtain the highest attention score. And the model selects high score words such as "cheap", "painted plastic", "nice" from the reviews accurately. From the color of reviews written by the sampled user/item, we can find that the model focuses on more informative reviews with the help of hierarchical attention. Compared with NARRE, we retain useful words in reviews. It seems that the attention mechanism works by giving weight scores to words and reviews according to their importance. That is to say, words and reviews with high scores can be selected for explanation recommendation.

Explanation Analysis of HANN

B00HG32UUW: You know, it's kind of pretty but definitely cheap. You get what you pay for, I guess. The ball at the end of the chain in the back arrived broken off but in the package. I put the bracelet on my wrist that I already had a magnetic bracelet on - turns out that THIS bracelet's not magnetic, not metal, and probably just painted plastic (Tibetan silver?) The same thing's true about the "turquoise" stone - probably just painted plastic. It's cheap but cute. I like it but probably wouldn't recommend it to anyone. You get what you pay for.

B00E1ZXX3G: Beautiful - I LOVE it! I've had a lot of compliments on it! - Very dainty, and fits nicely. Nice clasp, too!

B00BTD4I1E: I didn't want to get my seven-year-old grandson a digital watch, so I was excited to find this one at a great price. I gave this watch to him for Christmas, and he wears it all the time - and has learned to TELL time, too.

B0009KNC5Q: Very nice - exactly what I wanted - delicate, sterling silver, nice catch. I only wish I'd gotten another for my daughter.

Figure 3: Explanation Analysis of HANN

6 CONCLUSIONS

In this paper, we propose an approach for rating prediction using a hierarchical attention-based network named HANN, which can also distinguish the importance of reviews at both word level and review level automatically. Experiments on four real-life datasets from Amazon demonstrate that our model achieves an improvement in prediction compared to several state-of-the-art approaches. The hierarchical attention weights in sampled test data verify the effect on selecting informative words and reviews. As for future work, we will explore the potential advantages of the stochastic process for user dynamic preference modeling, which is worth studying that a user's preference may vary at different states in real scenarios. Besides, the same words at different aspect may reflect a different meaning based on its context.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable suggestions. This work was supported by the National Natural Science Foundation of China (NSFC) via grant 61632011, 61772153 and 71490722. This work was supported in part by Tencent Company. We would also like to thank Mingchen Yuan, Yang Wu and Zhaopeng Li for helpful comments.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. *international conference on learning representations* (2015).
[2] Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 2017. Aspect Based Recommendations: Recommending Items with the Most Valuable Aspects Based on User Reviews. (2017), 717–725.
[3] Peter Bühlmann and Sara Van De Geer. 2011. *Statistics for high-dimensional data: methods, theory and applications.* Springer Science & Business Media.
[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1583–1592.
[5] Hui Chen, Guiguang Ding, Zijia Lin, Sicheng Zhao, and Jungong Han. 2018. Show, Observe and Tell: Attribute-driven Attention Model for Image Captioning.. In

International Joint Conference on Artificial Intelligence. 606–612.

[6] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. 335–344. https://doi.org/10.1145/3077136.3080797

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[8] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 271–280.

[9] Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. 2012. From chatter to headlines: Harnessing the real-time web for personalized news recommendation. *WSDM 2012 - Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, 153–162. https://doi.org/10.1145/2124295.2124315

[10] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 193–202.

[11] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable Recommendation Through Attentive Multi-View Learning. https://www.microsoft.com/en-us/research/publication/explainable-recommendation-through-attentive-multi-view-learning/

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[13] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 507–517.

[14] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. TriRank: Review-aware Explainable Recommendation by Modeling Aspects. In *Conference on Information and Knowledge Management*. 1661–1670. https://doi.org/10.1145/2806416.2806504

[15] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 355–364.

[16] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer Product-based Neural Collaborative Filtering. In *International Joint Conference on Artificial Intelligence*. 2227–2233. https://doi.org/10.24963/ijcai.2018/308

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[18] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *international conference on learning representations* (2015).

[19] Michal Kompan and Mária Bieliková. 2010. Content-based news recommendation. In *International conference on electronic commerce and web technologies*. Springer, 61–72.

[20] Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101* (2017).

[21] 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.

[22] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*. 1532–1543.

[24] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten De Rijke. 2017. Social Collaborative Viewpoint Regression with Explainable Recommendations. (2017), 485–494.

[25] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 297–305.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[27] Xiang Wang, Xiangnan He, Liqiang Nie, and Tatseng Chua. 2017. Item Silk Road: Recommending Items from Information Domains to Social Users. *international acm sigir conference on research and development in information retrieval* (2017), 185–194.

[28] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tatseng Chua. 2017. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. *international joint conference on artificial intelligence* (2017), 3119–3125.

[29] Chenyan Xiong, Jimie Callan, and Tie-Yen Liu. 2017. Learning to attend and to rank with word-entity duets. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information, Tokyo, Japan*. 7–11.

[30] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J Smola, and Eduard H Hovy. 2016. Hierarchical Attention Networks for Document Classification. (2016), 1480–1489.

[31] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1449–1458.

[32] 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 & development in information retrieval*. ACM, 83–92.

[33] Sicheng Zhao, Guiguang Ding, Yue Gao, and Jungong Han. 2017. Approximating discrete probability distribution of image emotions by multi-modal features fusion. *Transfer* 1000, 1 (2017), 4669–4675.

[34] Sicheng Zhao, Guiguang Ding, Qingming Huang, Tat-Seng Chua, Björn W Schuller, and Kurt Keutzer. 2018. Affective Image Content Analysis: A Comprehensive Survey.. In *International Joint Conference on Artificial Intelligence*. 5534–5541.

[35] Sicheng Zhao, Yue Gao, Guiguang Ding, and Tat-Seng Chua. 2017. Real-time multimedia social event detection in microblog. *IEEE transactions on cybernetics* 99 (2017), 1–14.

[36] Sicheng Zhao, Hongxun Yao, Yue Gao, Guiguang Ding, and Tat-Seng Chua. 2016. Predicting personalized image emotion perceptions in social networks. *IEEE Transactions on Affective Computing* (2016).

[37] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 425–434.