Coevolutionary Recommendation Model: Mutual Learning between Ratings and Reviews

Conference Paper · April 2018

DOI: 10.1145/3178876.3186158

3 authors:

Yichao Lu
University of Toronto
5 PUBLICATIONS **44** CITATIONS
SEE PROFILE

Ruihai Dong
University College Dublin
28 PUBLICATIONS **243** CITATIONS
SEE PROFILE

Barry Smyth
University College Dublin
535 PUBLICATIONS **12,149** CITATIONS
SEE PROFILE

Coevolutionary Recommendation Model: Mutual Learning between Ratings and Reviews

Yichao Lu
University of Toronto
yichao@cs.toronto.edu

Ruihai Dong
Insight Centre for Data Analytics
University College Dublin
ruihai.dong@ucd.ie

Barry Smyth
Insight Centre for Data Analytics
University College Dublin
barry.smyth@ucd.ie

ABSTRACT

Collaborative filtering (CF) is a common recommendation approach that relies on user-item ratings. However, the natural sparsity of user-item rating data can be problematic in many domains and settings, limiting the ability to generate accurate predictions and effective recommendations. Moreover, in some CF approaches latent features are often used to represent users and items, which can lead to a lack of recommendation transparency and explainability. User-generated, customer reviews are now commonplace on many websites, providing users with an opportunity to convey their experiences and opinions of products and services. As such, these reviews have the potential to serve as a useful source of recommendation data, through capturing valuable sentiment information about particular product features. In this paper, we present a novel deep learning recommendation model, which co-learns user and item information from ratings and customer reviews, by optimizing matrix factorization and an attention-based GRU network. Using real-world datasets we show a significant improvement in recommendation performance, compared to a variety of alternatives. Furthermore, the approach is useful when it comes to assigning intuitive meanings to latent features to improve the transparency and explainability of recommender systems.

ACM Reference Format:
Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Coevolutionary Recommendation Model: Mutual Learning between Ratings and Reviews. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/https://doi.org/10.1145/3178876.3186158

1 INTRODUCTION

Recommender systems are an essential part of e-commerce platforms. They help customers to find what they are looking for and have been proven to drive sales and customer loyalty [17]. Collaborative Filtering (CF) [23] is a common recommendation approach that has been adopted by many e-commerce sites, from Netflix and Amazon to Digg and Zalando. Briefly, CF algorithms rely on user-item ratings, either directly [29] or indirectly (using latent factor models) [21], to make rating predictions and/or generate ranked recommendations. However, these approaches tend to suffer from the natural sparsity of the user-item ratings data; typically each user will only have "rated" a small fraction of the available products. Moreover, the latent features that are at the core of modern matrix factorization approaches [4] can lead to other problems, such as a lack of transparency and explainability.

1.1 User Reviews for Recommendation

The rise of user-generated reviews has introduced a novel source of recommendation data. Such reviews are plentiful and informative, and contain valuable information including the opinion of users on products and product features. For example, "After walking and biking all up and down the coast of Ambergris Caye, I am still positive that Caye Casa has the best restaurants and activities", tells us that the user likes the food in this restaurant, that walking and biking is a personal interest, and that the restaurant is close to the coast of Ambergris Cayes. Recently, such user reviews have been ustilized as the basis for new types of recommender systems. For example, [12] proposed a method to generate users and products profiles used in various recommendation tasks; see also [8–11]. And the ubiquitous nature of customer reviews makes them an important data source used to address the sparsity and transparency issues of CF algorithms.

Such techniques can be used to infer user ratings for products and services, and even combined with real ratings and more conventional ratings-based techniques, to generate improved recommendations; for example see [5]. One limitation of this type of approach is that it treats inferred ratings and real ratings as independent types of ratings data, combining their associated predictions/recommendations to generate final recommendations. Topic modeling methods such as Latent Dirichlet Allocation (LDA) [3] provide another way to integrate customer reviews into CF algorithms. For example, [39] proposed the method to combine latent feature based CF and probabilistic topic modeling to provide an interpretable latent structure for users and items; see also [7, 24]. The limitation of this approach is that it treats reviews as simple bags-of-words and, as such, ignores important sequential information that may help recommendation.

1.2 Deep Learning for Recommendation

Recently, Deep Learning has been adopted by recommender systems research, in part because of its ability to handle sequential information. For example, [37] proposed the method to treat a song as a set of 599 sequential frames, training a convolutional neural network (CNN) to learn its profile for the purpose of addressing the so-called cold-start problem [31] in recommendation. A customer review can also be considered as a sequence of words. For example, [19] proposed a method that integrates a product description summarized by a trained CNN network into probabilistic matrix

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/https://doi.org/10.1145/3178876.3186158

factorization. Compared to topic models, it is capable of catching the contextual information of a document.

In related work, Recurrent Neural Networks (RNN) have been applied to various natural language processing tasks with great success. For instance, Bidirectional RNN, including Bidirectional Long Short-Term Memory (Bi-LSTM) [14], and Bidirectional Gated Recurrent Unit (Bi-GRU) [43], can encode a target word with contextual and sequential information (encoding not just the target word but also the surrounding words), when using sentences, documents as input sequences. RNNs are commonly used to capture or summarize the meaning of sentences or documents in various tasks such as machine translation[2], sentence summarization [30] and sentiment analysis [36]. Meanwhile, attention-based methods, based on the visual attention mechanism found in humans, by learning weight vectors for different sub-tasks, are becoming widely used in machine translation [6] and image tracking [42]. Similar ideas have also proven helpful for extracting textual features from customer reviews; see for example [33].

1.3 Main Contributions

Inspired by the recent success of attention-based models and RNNs, in this paper we propose an attention-based mechanism to learn representative features from user-generated reviews and combine this with a conventional matrix factorization recommendation model as shown in Figure 1.

The contribution of this paper is threefold:

(i) We introduce a novel recommendation model called *TARMF*, which utilizes attention-based recurrent neural networks to extract topical information from review documents.
(ii) We demonstrate how textual features can be applied to enhance the performance of matrix factorization recommendation techniques, and propose an optimization algorithm for training our *TARMF* model.
(iii) We demonstrate the ability for *TARMF* to achieve superior recommendation performance on five publicly available benchmark datasets, in comparison to a variety of state-of-the-art baseline alternatives.

2 RELATED WORK

2.1 Latent Factor Models in Recommender Systems

The latent factor models are a set of collaborative filtering approaches widely used in the literature of recommender systems. Through characterizing each user and item as a fixed dimension vector, latent factor models could learn user preferences and item features from observed ratings, and accordingly recommend new items to users.

Among all the different alternatives of latent factor models, matrix factorization based methods [20, 21] are arguably the most prevalent ones. Essentially, matrix factorization turns the recommendation task into a matrix completion problem. To date much of the state-of-the-art recommendation models are built upon matrix factorization techniques. For example, the Probabilistic Matrix Factorization (PMF) model [27] is a widely adopted framework with reliable performance.

The problem with pure matrix factorization models is that, the number of unobserved ratings scales linearly with the product of the number of users and the number of items, while the number of known ratings typically scales linearly with the number of users. Therefore with the rapid growth of user numbers in modern e-commerce platforms, the increasing sparsity of the data becomes a critical concern.

One way to solve the sparseness problem is to mine useful features from user-generated content, e.g., user reviews, movie plots, and item usage instructions. For example, [7, 24, 39] proposed to use topic modeling to learn features from review documents, based on which matrix factorization techniques could gain useful prior knowledge of the distribution of parameters.

In addition, utilizing user-generated content in latent factor models helps to improve the interpretability of recommendations. In [24], the authors demonstrate that displaying the top-k words of an LDA model could yield meaningful word clusters related to distinct topics. [44] explores the effectiveness of explicitly aligning the latent factors and review aspects, which results in an explainable model that could make reasoning about recommendation choices.

2.2 Deep Neural Networks in Natural Language Processing

The recent enthusiasm for applying deep learning techniques in natural language processing originates from the success of learning representative word vectors [26, 28]. With the utilization of meaningful word embedding vectors, almost all deep computational frameworks used in the literature of computer vision and speech recognition can be seamlessly applied to natural language processing.

Most widely used neural network structures, including convolutional neural networks [16, 19], recurrent neural networks [25, 35], and neural turing machine [41], have shown promising results in various natural language processing benchmarks. Specifically, the attention mechanism introduced by [2] enabled neural language models to achieve state-of-the-art results in machine translation [13, 38], reading comprehension [15, 32], speech recognition [1], etc.

The success of deep neural networks in a variety of natural language processing tasks has raised the attention of the recommender systems community as well. For example, [18] proposed to employ a convolutional neural network to facilitate the learning of matrix factorization. Similarly, [33] utilized an attention-based convolutional neural network for modeling review documents, and achieved state-of-the-art results in the task of rating prediction.

3 TOPICAL ATTENTION REGULARIZED MATRIX FACTORIZATION

In this section, we present the detail of our proposed model, Topical Attention Regularized Matrix Factorization (*TARMF*). We begin by describing the attention-based recurrent neural network architecture we utilized for document modeling, followed by the approach of extracting textual features from user and item review documents. We then introduce an extension of the traditional probabilistic matrix factorization model by incorporating textual regularization.

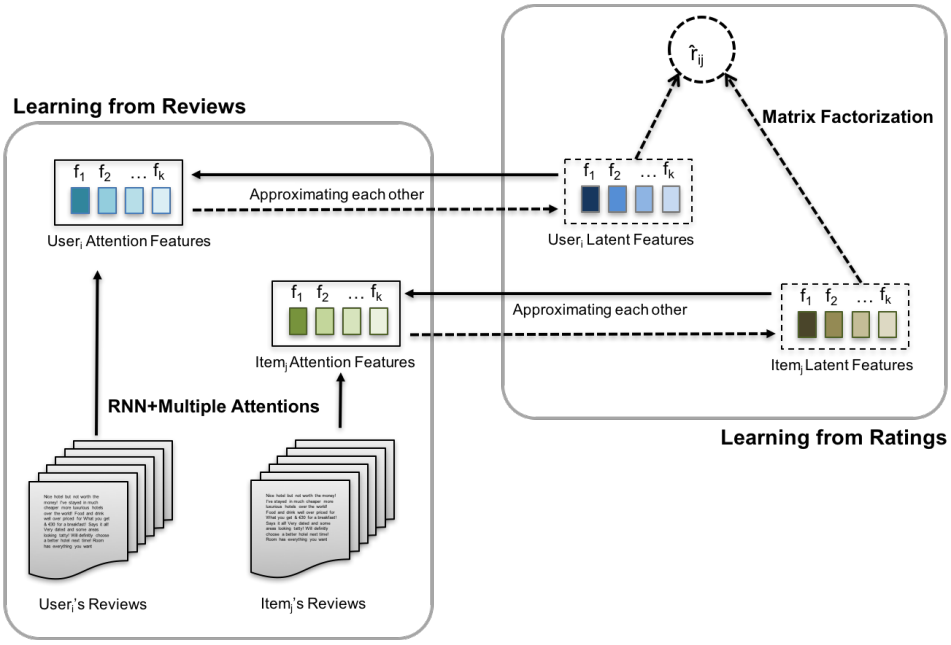


Figure 1: High-level architecture of mutual learning between reviews and ratings. Textual features of user_i and item_j are extracted from their review documents by utilizing bidirectional recurrent neural networks with topical attention mechanism. Latent features are extracted from the matrix factorization model. Textural features and latent features approximate to each other during training.

Finally, we present a computational framework for optimizing the parameters.

3.1 Attention-Based Recurrent Neural Network for Document Modeling

We employ a bidirectional recurrent neural network with attention mechanism to learn representative features from review documents. The network architecture consists of four primary components: (i) a word embedding layer, (ii) a sequence encoding layer, (iii) a topical attention layer, and (iv) a feature projection layer; see Figure 2.

3.1.1 Word Embedding Layer. The word embedding layer takes as input a sequence of words $(w_1, w_2, w_3, ..., w_T)$, and maps each word to its respective k-dimensional vector representation $x_i \in \mathcal{R}^k$. The vector representations are expected to encode the semantic and syntactic information carried by each word, thus enabling the sequence encoding layer to effectively capture the contextual dependencies of the input sequence. We initialize the word embedding layer with pre-trained word vectors obtained from *word2vec* [26], and then fine-tune it with back-propagation.

3.1.2 Sequence Encoding Layer. The sequence encoding layer provides contextual annotations for the input sequence. Specifically, we utilize the bidirectional GRU architecture proposed by [6] due to its computational efficiency and robust performance in our experiments.

A Gated Recurrent Unit (GRU) is a popular variant of the vanilla recurrent hidden unit. Through utilizing gating units to modulate the flow of information, each recurrent unit is capable of encapsulating sequential dependencies across different time scales.

Formally speaking, a GRU computes its activation at time step t as the linear interpolation between the previous activation h_{t-1} and the candidate activation $\widetilde{h_t}$:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t}, \tag{1}$$

where

$$\widetilde{h_t} = tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h), \tag{2}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \tag{3}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_z). \tag{4}$$

The *update gate* z_t decides the extent to which past information is superseded by new information, while the *reset gate* r_t determines the degree that the previous activation contributes to the candidate activation.

In order for the annotations to summarize the information from both the preceding words and the following words, we employ a bidirectional GRU consisting of forward and backward GRUs. The forward GRU reads the input sequence in the usual order, while the backward GRU reads the input sequence in the reversed order. The activations of the forward GRU and the backward GRU at time step t are denoted as $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, respectively. At each time step, we concatenate the forward activation and the backward activation to obtain the final annotation, i.e., $h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right]$.

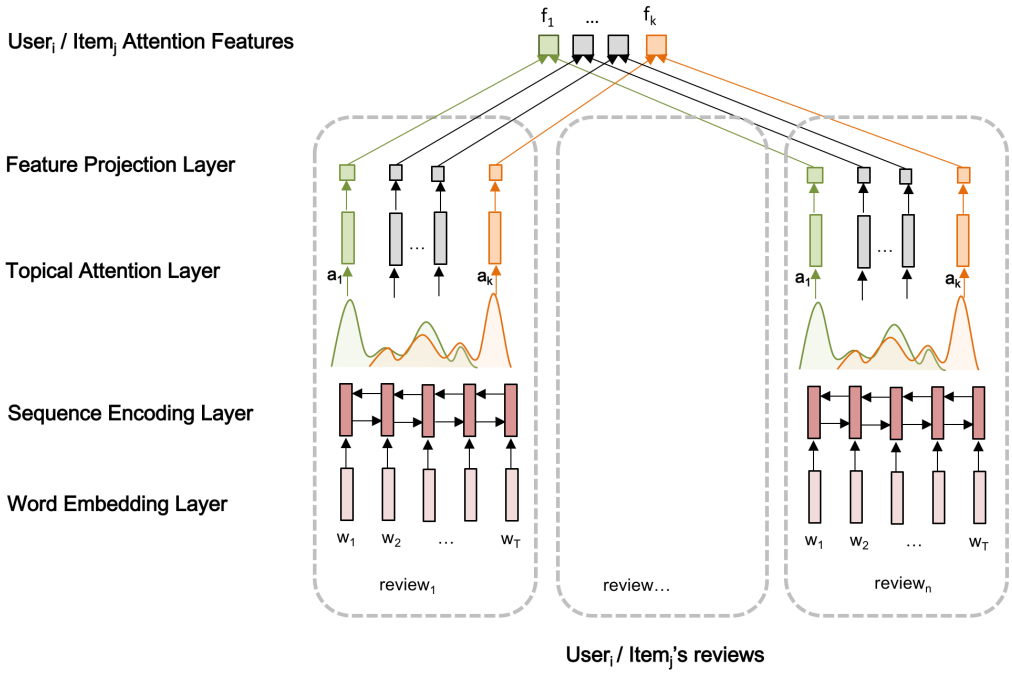


Figure 2: The attention-based bidirectional GRU network for document modeling.

3.1.3 Topical Attention Layer. The topical attention layer extracts topic-related information associated with the set of topics of interest to the recommendation task. We assume that not all parts of a document are equally relevant to a specific topic. Therefore we introduce the attention mechanism to capture the relative importance between different words.

Suppose that each user and item can be characterized by the corresponding K-dimensional latent factor vector, each latent dimension is expected to represent a topic related to the specific user or item. Intuitively, the distribution of attention weights on each word should be different for each topic. Thus we employ K distinct attention modules corresponding to the K topics.

Consider, for example, the k-th attention module. Given the sequence of word annotations $(h_1, h_2, h_3, ..., h_T)$, the attention module first transforms each word annotation through a single layer perceptron with the *tanh* activation function:

$$s_t^k = tanh(W_s^k h_t + b_s^k). \tag{5}$$

Then the attention module compares the similarities between a context vector z_k and the transformed annotations by computing the dot products, and assigns each annotation a weighting score with the **softmax** function:

$$a_t^k = \frac{z_k \cdot s_t^k}{\sum_{t=1}^{T} z_k \cdot s_t^k}. \tag{6}$$

Finally, the attention module computes its output $\hat{h}_k$ as the weighted sum of the annotations:

$$\hat{h}_k = \sum_{t=1}^{T} a_t^k h_t. \tag{7}$$

The output of each individual attention module is passed together to the feature projection layer as the activation of the topical attention layer.

3.1.4 Feature Projection Layer. The feature projection layer performs non-linear transformations on the feature representations generated by the penultimate layer. We employ a single layer perceptron with *tanh* as its activation function. The activation for the k-th attention module is thus transformed as:

$$c_k = tanh(W_c^k \hat{h}_k + b_c^k). \tag{8}$$

The feature projection layer concatenates the transformed activations, and outputs it as the latent document representation, i.e., $c = [c_1, c_2, c_3, ..., c_K]$.

3.2 Extracting Textual Features from Review Documents

We assume that the textual features extracted from review documents can serve as a reasonable indicator of the user and item latent factor vectors. To begin with, we need to define the concept of a *review document*. We define the user review document $d_{u,i}$ as the set of all reviews written by user i. Similarly, the item review document $d_{v,j}$ is defined to be the collection of reviews written on item j.

Note that the review written by user i on item j would be included both in the user review document $d_{u,i}$ and the item review

document $d_{v,j}$. However, the same review should be treated differently in these circumstances. For reviews in the user review document, we expect to learn user preferences revealed in the content. When it comes to the reviews in the item review document, our aim is to extract the features related to the specific item. Due to the inherent difference of the user review document and the item review document, they are modeled by two attention-based recurrent neural networks with the same architecture and different parameters. The attention-based recurrent neural networks for modeling user review documents and item review documents are named as the user attention network and the item attention network, respectively.

Given a review document, we first generate the latent document representation for each individual review with the attention-based recurrent neural network, and then average them as the textual features extracted from the review document.

3.3 Textual Regularized Matrix Factorization

We extend the Probabilistic Matrix Factorization (PMF) model [27] by introducing textual regularization in the user and item latent factor vectors.

Suppose we have N users and M items, matrix factorization finds a user coefficient matrix $U \in \mathcal{R}^{D\times N}$ and an item factor matrix $V \in \mathcal{R}^{D\times M}$ whose product $\hat{R} = U^T V$ approximates the rating matrix $R \in \mathcal{R}^{N\times M}$. The column vectors u_i and v_j are the D-dimensional distributed representations for user i and item j, respectively. For a linear model with Gaussian observation noise, the conditional distribution over the observed ratings can be defined as in Equation 9, where $\mathcal{N}(\mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2, and I_{ij} is an indicator function where $I_{ij} = 1$ if user i rated item j and $I_{ij} = 0$ otherwise.

$$p(R \mid U, V, \sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij} \mid u_i^T \cdot v_j, \sigma^2)\right]^{I_{ij}} \tag{9}$$

Unlike the traditional probabilistic matrix factorization model, which places zero-mean isotropic Gaussian prior distributions on all the latent variables, the prior means of the user and item latent factors in the *TARMF* model are not fixed at zero. Instead, we assume that the user and item latent factors are highly correlated with the textual features extracted from the review documents. Therefore, we define the prior distributions of the user and item latent factor vectors as:

$$p(U|\widetilde{U}, \sigma_U^2) = \prod_{i}^{N}\mathcal{N}(U_i|\widetilde{U}_i, \sigma_U^2 I), \tag{10}$$

$$p(V|\widetilde{V}, \sigma_V^2) = \prod_{j}^{M}\mathcal{N}(V_j|\widetilde{V}_j, \sigma_V^2 I), \tag{11}$$

where $\widetilde{U}_i$ and $\widetilde{V}_j$ are the textual features extracted from the review documents of user i and item j, as described in Section 3.2.

The introduction of the textual features in the prior distributions essentially regularizes the matrix factorization model so that it could generalize well on the unseen test dataset.

3.4 Optimization Methodology

Training the *TARMF* model involves optimizing the following unknown parameters: (i) the user coefficient matrix U, (ii) the item factor matrix V, (iii) the parameters W_U in the user attention network, and (iv) the parameters W_V in the item attention network.

While the optimization objective of U and V is straightforward, i.e., to minimize the difference between the rating matrix R and the product of U and V, the optimization criterion of W_U and W_V is still unclear. Since we expect the textual features to serve as reliable indicators of the latent factor vectors, $\widetilde{U}_i$ and $\widetilde{V}_j$ should approximate U_i and V_j. We therefore optimize W_U and W_V through maximizing $sim(U, \widetilde{U})$ and $sim(V, \widetilde{V})$, where the sim function measures the similarity between two matrices.

An intuitive optimization strategy is to define an overall loss function, and to train all the parameters simultaneously with stochastic gradient descent. Nevertheless, due to the high correlation between the parameters, the stochastic gradient descent algorithm could easily get trapped in one of the undesired local minima. Consider, for example, the user coefficient matrix U and the parameters W_U of the user attention network. On one hand, U is dependent upon W_U as changes in W_U would affect the textual features $\widetilde{U}$ generated by the user attention network, and thus altering the posterior distribution of U. On the other hand, W_U is optimized through maximizing the similarity between U and $\widetilde{U}$. Thus, when U and W_U are jointly optimized, they are likely to mislead each other which results in the deterioration of model performance. The same situation holds for the optimization of V and W_V as well.

Instead of jointly optimizing all the unknown parameters, we adopt an alternative approach, which iteratively updates each of the four sets of parameters in a specific order. When the model is optimizing a particular set of parameters, we temporarily fix all the remaining parameters to be constant. Our rationale is that, such iterative training methodology can help to alleviate the dependency between the parameters, and accordingly facilitate the training process.

Suppose that the optimal $\widetilde{U}$ and $\widetilde{V}$ are known and fixed, the posterior distribution over U and V is given by

$$\begin{aligned}&\max_{U,V} p(U, V|R, \widetilde{U}, \widetilde{V}, \sigma^2, \sigma_U^2, \sigma_V^2)\\ =&\max_{U,V} p(R|U, V, \sigma^2)p(U, V|\widetilde{U}, \widetilde{V}, \sigma_U^2, \sigma_V^2),\end{aligned} \tag{12}$$

where

$$p(U, V|\widetilde{U}, \widetilde{V}, \sigma_U^2, \sigma_V^2) = p(U|\widetilde{U}, \sigma_U^2)p(V|\widetilde{V}, \sigma_V^2) \tag{13}$$

is the joint posterior distribution of U and V given $\widetilde{U}$, $\widetilde{V}$, σ_U^2, and σ_V^2.

Maximizing the posterior probability is equivalent to minimizing its negative logarithm, which is given by

$$\begin{aligned}\mathcal{L}(U, V|R, \widetilde{U}, \widetilde{V}) = &\frac{1}{2}\sum_{i}^{N}\sum_{j}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2\\ &+ \frac{\lambda_U}{2}\sum_{i}^{N}\|U - \widetilde{U}\|_F^2 + \frac{\lambda_V}{2}\sum_{i}^{N}\|V - \widetilde{V}\|_F^2,\end{aligned} \tag{14}$$

where $\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_V = \sigma^2/\sigma_V^2$, and $\|\cdot\|_F$ denotes the Frobenius norm.

Note that Equation 14 becomes a quadratic function with respect to U (or V) when V (or U) is treated as constant, which implies that the equation reaches its optimal solution when the gradient of U (or V) equals zero. Therefore we adopt the alternating least squares technique which repeatedly optimizes one of U and V while temporarily fixing the other to be constant:

$$U_i = (VI_iV^T + \lambda_U I_K)^{-1}(VR_i + \lambda_U \widetilde{U_i}), \tag{15}$$

$$V_j = (UI_jU^T + \lambda_V I_K)^{-1}(UR_j + \lambda_V \widetilde{V_j}), \tag{16}$$

where $I_i \in \mathcal{R}^{M\times M}$ is a diagonal matrix with I_{ij} as its diagonal elements, and $R_i \in \mathcal{R}^M$ is a vector of R_{ij}. Recall that $I_{ij} = R_{ij} = 0$ if user i has not yet rated item j. I_j and R_j are defined in an analogous manner.

Conversely, consider the circumstance in which the optimal U and V are known a priori. The goal of the user and item attention network is then to adjust their internal weights W_U and W_V so that the textual features they extract could approximate the ideal U and V. For a given user i with user review document $X_{u,i}$ and user latent factor U_i, we can define the loss function for W_U as

$$\mathcal{L}_{\mathcal{W}_u}(X_{u,i}, U_i) = \|UAN(W_U, X_{u,i}) - U_i\|_F^2, \tag{17}$$

where $\widetilde{U_i} = UAN(W_U, X_{u,i})$ denotes the textual features for user i generated by feeding the user review document $X_{u,i}$ into the user attention network with parameters W_U.

The loss function for W_V can be similarly defined as follows:

$$\mathcal{L}_{\mathcal{W}_v}(X_{v,j}, V_j) = \|IAN(X_{v,j}) - V_j\|_F^2, \tag{18}$$

where IAN refers to the item attention network.

The full algorithm for optimizing the *TARMF* model is presented in Algorithm 1. At each epoch, we alternate between the optimization of U, V, W_U, and W_V. While U and V are fitted with alternating least squares, W_U and W_V are optimized with mini-batch gradient descent. The parameters currently being optimized make the assumption that all the other parameters are optimal. Apparently, such assumption is far from the truth in the first few epochs, and the parameters might be falsely guided by other unoptimized parameters. However, as the model goes through the optimization procedure, each parameter is getting closer and closer to its optimal value, and the model would eventually converge.

4 QUANTITATIVE EVALUATION

In this section, we evaluate the *TARMF* model on real-world datasets to compare its performance with a number of state-of-the-art recommendation techniques reported in the literature.

4.1 Datasets and Evaluation Metrics

We use five publicly available datasets - including two datasets from the Yelp Dataset Challenge [1] and three others from Amazon - for the purpose of this analysis; see Table 1.

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

Algorithm 1: Optimization Algorithm for *TARMF*

Randomly initialize the user coefficient matrix $U \in R^{K\times N}$.
Randomly initialize the item factor matrix $V \in R^{K\times M}$.
Initialize the parameters W_U in the user attention network.
Initialize the parameters W_V in the item attention network.
for *epoch* $\leftarrow 1$ **to** T **do**
  for $i \leftarrow 1$ **to** N **do**
    Update U_i with least square approximation:
    $$U_i \leftarrow (VI_iV^T + \lambda_U I_K)^{-1}(VR_i + \lambda_U \widetilde{U_i})$$
  end
  for $j \leftarrow 1$ **to** M **do**
    Update V_j with least square approximation:
    $$V_j \leftarrow (UI_jU^T + \lambda_V I_K)^{-1}(UR_j + \lambda_V \widetilde{V_j})$$
  end
  for *iteration* $\leftarrow 1$ **to** S **do**
    Randomly sample a mini-batch of users X_U.
    Update W_U via stochastic gradient descent:
    $$W_U \leftarrow W_U - \eta \frac{\partial \mathcal{L}_{\mathcal{W}_u}(X_U)}{\partial W_U}$$
    Randomly sample a mini-batch of items X_V.
    Update W_V via stochastic gradient descent:
    $$W_V \leftarrow W_V - \eta \frac{\partial \mathcal{L}_{\mathcal{W}_v}(X_V)}{\partial W_V}$$
  end
end

Dataset	#users	#items	#ratings
Yelp 2013	1,631	1,633	78,966
Yelp 2014	4,818	4,194	231,163
Amazon Electronics	37,128	25,783	1,689,188
Amazon Video Games	24,303	10,672	231,780
Amazon Gourmet Foods	14,681	8,713	151,254

Table 1: Statistics of the evaluation datasets

We first randomly split all the five datasets into training / validation / test sets with a 70 /10 / 20 split. We then tune the hyperparameters on the validation set, and evaluate the performance of different approaches by calculating the Mean Squared Error (MSE) on the test set, which compares the differences between the predicted ratings and the golden truth:

$$MSE = \frac{\sum_{i=1}^{N}(r_i - \hat{r}_i)^2}{N} \tag{19}$$

4.2 Baseline Models

For the purpose of comparison, we examine the performance of the *TARMF* model together with the following baseline models:

(i) **Offset**: The offset estimator takes the average across all the ratings in the training set, and utilizes it as the predictions of the ratings in the test set.

(ii) **PMF**: Probabilistic Matrix Factorization (PMF) [27] is a popular factor-based model from a probabilistic point of view that performs well on very sparse and imbalanced datasets.
(iii) **HFT**: Hidden Factor as Topics (HFT) [24] is a novel recommendation technique that utilizes Latent Dirichlet Allocation [3] to model review documents. The model is optimized through considering both the errors in the predicted ratings and the corpus likelihood of the learned latent factors.
(iv) **CTR**: Collaborative Topic Regression (CTR) [39] learns interpretable latent structure from user generated content so that probabilistic topic modeling can be integrated into collaborative filtering.
(v) **JMARS**: Jointly Modeling Aspects, Ratings, and Sentiments (JMARS) [7] is another state-of-the-art probabilistic model that combines collaborative filtering and topic modeling.
(vi) **ConvMF+**: Convolutional Matrix Factorization (ConvMF) [18] is a newly proposed recommendation model that employs a convolutional neural network for learning item features from item review documents. ConvMF+ refers to the ConvMF model initialized with pre-trained word embeddings.

4.3 Tuning Hyperparameters

We explore how different settings of the hyperparameters would influence the performance of our proposed model. The examined hyperparameters include the dimension of the word embeddings d_W, the dimension of the sequence encoder d_S, the dimension of the transformed annotations in the attention module d_A, and the regularization terms λ_U and λ_V.

The validation MSE as a result of varying d_W, d_S, d_A, λ_U, and λ_V are presented in Figure 3, Figure 4, Figure 5, and Figure 6, respectively. As we can see, the optimal value of each hyperparameter remains the same regardless of the evaluated dataset. Therefore we empirically set the word embedding dimension to be 128, the sequence encoder state dimension to be 64, the dimension of the transformed annotations in the attention module to be 64, and λ_U and λ_V to be 100.

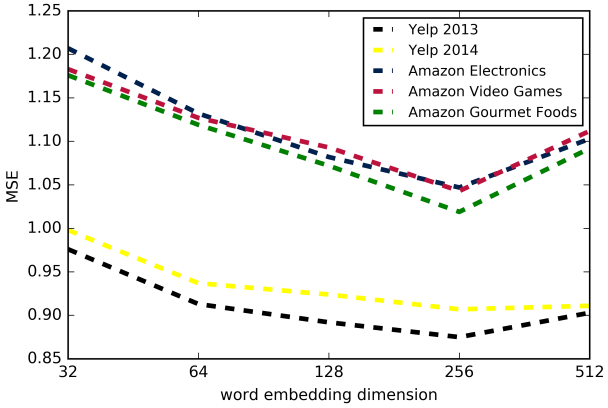


Figure 3: Validation MSE as a result of varying d_W.

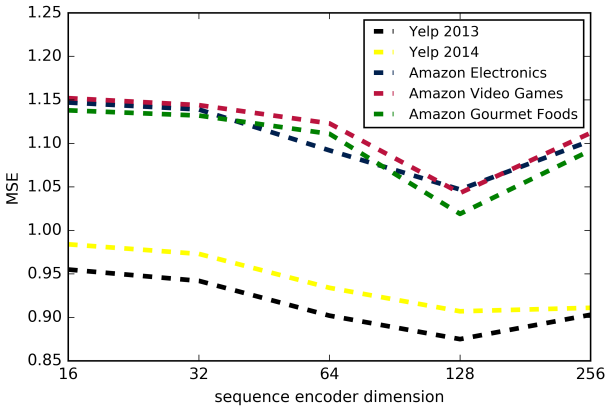


Figure 4: Validation MSE as a result of varying d_S.

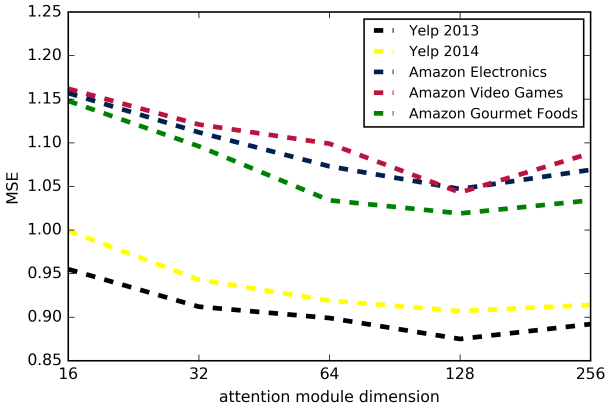


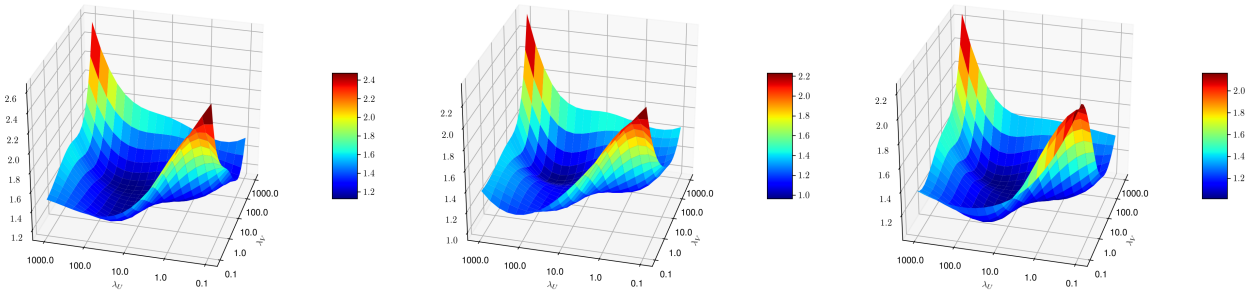
Figure 5: Validation MSE as a result of varying d_A.

4.4 Evaluation Results

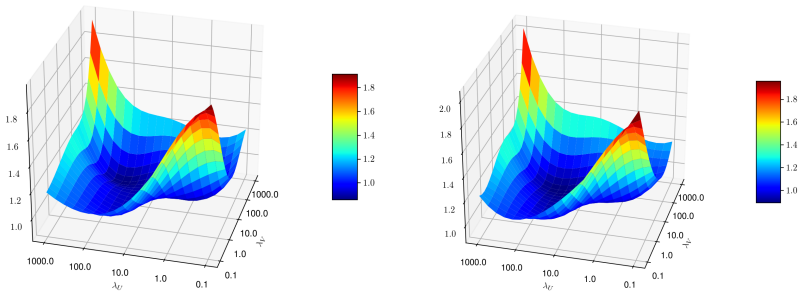
The evaluation results of all the compared models are presented in Table 2. We note that the *TARMF* model outperforms all the baseline models across all the five datasets. Furthermore, these differences between our proposed method and each of the baseline models are statistically significant for $p < 0.05$.

The evaluation results actually meet our expectations. The offset estimator has the poorest performance as it makes constant predictions regardless of the difference of users and items. The PMF model, on the other hand, characterizes users and items with latent factors. Nevertheless, building such model completely from ratings can be quite hard, especially when the rating data is sparse. Therefore the PMF model still cannot yield satisfying results.

The remaining five algorithms, i.e., HFT, CTR, JMARS, ConvMF+, and *TARMF*, all utilize user-generated content as an auxiliary source of information for recommendation. This has proven to be an effective approach to deal with the sparseness problem. In particular,



(a) Amazon Electronics (b) Amazon Video Games (c) Amazon Gourmet Foods



(d) Yelp 2013 (e) Yelp 2014

Figure 6: Validation MSE as a result of varying λ_U and λ_V.

	Dataset				
Model	Yelp 2013	Yelp 2014	Amazon Electronics	Amazon Video Games	Amazon Gourmet Foods
Offset	1.017	1.125	1.476	1.435	1.397
PMF	0.985	1.053	1.411	1.297	1.251
HFT	0.977	1.029	1.259	1.152	1.121
CTR	0.975	1.013	1.284	1.147	1.139
JMARS	0.970	0.998	1.244	1.133	1.114
ConvMF+	0.917	0.954	1.241	1.092	1.084
TARMF	**0.875**	**0.909**	**1.147**	**1.043**	**1.019**

Table 2: Recommendation performance in terms of MSE

HFT, CTR, and JMARS are based on topic modeling with the bag-of-words assumption. Due to the inherent limitation of the bag-of-words model, i.e., it completely ignores the context of each word, these model are not fully capable of capturing the textual features in the review document. The ConvMF+ model, which employs a convolutional neural network for document modeling, partially solved this problem by integrating a set of filters corresponding to n-gram features in the text. It therefore significantly outperforms the bag-of-words models.

The *TARMF* model is the best performing algorithm among all the five compared models. Similar to the ConvMF+ model, the *TARMF* model relaxes the bag-of-words assumption as well. In addition, it further improves the ConvMF+ model in three approaches. Firstly, the *TARMF* model employs a bidirectional recurrent neural network with attention mechanism, which is capable of modeling long documents. Secondly, the *TARMF* model applies the topical attention approach, which is similar to the idea of topic modeling, so that each latent factor dimension can be aligned with a specific topic. And thirdly, the ConvMF+ model only considers the item review documents, while the *TARMF* model takes both the user and item review documents into accounts, which introduces more flexibility to the model.

5 QUALITATIVE EVALUATION

5.1 Attention Visualization

In order to understand the mechanism behind the recommendation, we try to visualize the positions that each attention module attend

to. Consider a particular word in a review text of length T, the expected attention score assigned by each attention module is $1/T$. We assume that if a word w_t is assigned with an attention score $s_t \geq 5/T$ by a specific attention module, the word is then of interest to the particular attention module. In addition, if a word simultaneously reaches the attention threshold of different attention modules, we assume that it is only attended by the attention module that assigns it the maximum attention score.

Figure 7 and Figure 8 visualize the attention distribution of a specific review in the amazon electronics dataset assigned by the user and item attention networks. Words attended by different attention modules are highlighted with distinct colors, and darker colors refer to higher attention scores. We can make several observations from these figures. Firstly, the attention modules can learn interpretable regions of interest in the review text. For example, the red highlights in Figure 7 corresponding to the first attention module, extracts information about children. Through analyzing these information, the model could learn that the user that wrote this review has interests in purchasing electronic devices for children. Similarly, the yellow highlights represent the responding speed of the device, while the blue highlights refers to the price. Secondly, the attention distributions assigned by the user and item attention network are nicely aligned. For example, the yellow highlights in both Figure 7 and Figure 8 consider the responding speed of the device. Such attention alignment is crucial as the rating is predicted by the dot product between the user and item latent factor vectors.

I bought this ebook (16G) for my kindergarten and elementary children to read books on trips and my old child to check emails. It does the jobs well until now. I personally like it very much for its excellent hardware performance. With low price, fast response, and light weight, book size, and Barn and Nobles support, It is the best device for children when you want to have something as an alternative for your computer. I am a fan of Amazon.com and meant to buy a kindle for my children. But the displayed sample on my local Bestbuy store showed me that the nook tablet responded much faster than kindle fire. It could not be generally true, but based on the displayed tablets, I had to choose the nook. Until now, I have had mainly happy experiences with it. It has apps for children, one of them has many Smithsonian videos that my small children love the most.

Figure 7: Attention visualization of user attention network.

I bought this ebook (16G) for my kindergarten and elementary children to read books on trips and my old child to check emails. It does the jobs well until now. I personally like it very much for its excellent hardware performance. With low price, fast response, and light weight, book size, and Barn and Nobles support, It is the best device for children when you want to have something as an alternative for your computer. I am a fan of Amazon.com and meant to buy a kindle for my children. But the displayed sample on my local Bestbuy store showed me that the nook tablet responded much faster than kindle fire. It could not be generally true, but based on the displayed tablets, I had to choose the nook. Until now, I have had mainly happy experiences with it. It has apps for children, one of them has many Smithsonian videos that my small children love the most.

Figure 8: Attention visualization of item attention network.

5.2 Discussion

It is straightforward to see that the attention-based GRU network has indeed learned to selectively attend to content of interest when modeling different topics in a review text. We can gain intuitive interpretation of the hidden topic related to each dimension of the latent factor by examining regions with high attention scores. One would expect that we could automatically generate such interpretations by gathering words with high attention scores. For example, in [24], the authors demonstrate that extracting the top k words of the LDA model would yield explainable topics. However, this approach does not work for the purpose of this paper.

The reason is that, instead of making the bag-of-words assumption as in LDA, the GRU network incorporates sequential information in its annotations. Therefore, the annotation of each word does not only contain its own semantic meaning, but also include information from its surrounding words. As a consequence, the attention scores no longer measure the importance of the word alone. In fact, the attention scores summarize the level of interest of the context. We can see in Figure 8 that, in "I am a fan of Amazon.com and meant to buy a kindle for my children. but the displayed sample on my local Bestbuy store showed me that the nook tablet responded much faster than kindle fire", instead of attending to "children", the item attention network has actually attended to the "for" in front of it. Therefore we cannot directly generate recommendation explanations by extracting words with high attention scores in each topic. We leave the automatic creation of recommendation interpretations as a future work.

6 CONCLUSION AND FUTURE WORK

Incorporating textual features has proven to enhance the performance of collaborative filtering algorithms. In this paper, we present a novel idea that employs an attention-based GRU network to facilitate matrix factorization, and introduce a coevolutionary algorithm for optimizing the recommendation model. The proposed model, which we name as *TARMF*, achieves state-of-the-art results on all of the five benchmark datasets. In addition, we demonstrate how the attention weights assigned by each attention module can be utilized to interpret the meaning associated with each dimension of the latent factor vectors. However, simply identifying the representative words for each hidden topic is far from enough for providing personalized recommendation explanations. The desirable accompanying explanation for each recommendation should be written in informative and readable human languages. Meanwhile, the recent success of applying deep neural networks and deep reinforcement learning algorithms for natural language generation [22, 34, 40] has made it feasible to create meaningful and relevant texts conditioned on specific topics. Accordingly, such natural language generation models can be employed for explaining the suggestions provided by recommender systems, through learning to transform the learned latent features into human readable texts. Therefore, in future work, we will explore the potential of utilizing a sequence-to-sequence (seq2seq) learning framework [35] for the purpose of generating persuasive recommendation explanations that can help customers make better purchasing decisions.

ACKNOWLEDGMENTS

This work is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289.

REFERENCES

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*. 173–182.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

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

[4] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9, 6 (2009), 717.

[5] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction* 25, 2 (2015), 99–154.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[7] 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.

[8] Ruihai Dong, Michael P O'Mahony, Markus Schaal, Kevin McCarthy, and Barry Smyth. 2013. Sentimental product recommendation. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 411–414.

[9] Ruihai Dong, Michael P O'Mahony, and Barry Smyth. 2014. Further experiments in opinionated product recommendation. In *International Conference on Case-Based Reasoning*. Springer, 110–124.

[10] Ruihai Dong, Markus Schaal, Michael P O'Mahony, Kevin McCarthy, and Barry Smyth. 2013. Opinionated product recommendation. In *International Conference on Case-Based Reasoning*. Springer, 44–58.

[11] Ruihai Dong, Markus Schaal, Michael P O'Mahony, and Barry Smyth. 2013. Topic Extraction from Online Reviews for Classification and Recommendation.. In *IJCAI*, Vol. 13. 1310–1316.

[12] Ruihai Dong and Barry Smyth. 2016. Personalized Opinion-Based Recommendation. In *International Conference on Case-Based Reasoning*. Springer, 93–107.

[13] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122* (2017).

[14] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5-6 (2005), 602–610.

[15] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*. 1693–1701.

[16] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).

[17] Timothy L Keiningham, Bruce Cooil, Lerzan Aksoy, Tor W Andreassen, and Jay Weiner. 2007. The value of different customer satisfaction and loyalty metrics in predicting customer retention, recommendation, and share-of-wallet. *Managing Service Quality: An International Journal* 17, 4 (2007), 361–384.

[18] 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, 233–240.

[19] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[20] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

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

[22] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541* (2016).

[23] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[24] 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.

[25] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

[26] 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.

[27] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[28] 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 (EMNLP)*. 1532–1543.

[29] Alexandrin Popescul, David M Pennock, and Steve Lawrence. 2001. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 437–444.

[30] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* (2015).

[31] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 253–260.

[32] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016).

[33] 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.

[34] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714* (2015).

[35] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[36] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification.. In *EMNLP*. 1422–1432.

[37] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in neural information processing systems*. 2643–2651.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv preprint arXiv:1706.03762* (2017).

[39] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[40] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745* (2015).

[41] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698* (2015).

[42] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*. 2048–2057.

[43] 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*. 1480–1489.

[44] 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.