Applied Soft Computing Journal 100 (2021) 106985

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Deep neural review text interaction for recommendation systems

Parisa Abolfath Beygi Dezfouli, Saeedeh Momtazi *, Mehdi Dehghan

Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 14 March 2020
Received in revised form 9 October 2020
Accepted 1 December 2020
Available online 9 December 2020

Keywords:
Recommender systems
Rating prediction
Neural text similarity
Review processing
Convolutional neural networks

ABSTRACT

Users' reviews contain valuable information which are not taken into account in most recommender systems. According to the latest studies in this field, using review texts could not only improve the performance of recommendation, but it can also alleviate the impact of data sparsity and help to tackle this problem. In this paper, we present a neural recommender model which recommends items by leveraging user reviews. In order to predict user rating for each item, our proposed model, named *MatchPyramid Recommender System* (MPRS), represents each user and item with their corresponding review texts. Thus, the problem of recommendation is viewed as a text matching problem such that the matching score obtained from matching user and item texts could be considered as a good representative of their joint extent of similarity. To solve the text matching problem, inspired by MatchPyramid (Pang et al., 2016), we employed an interaction-based approach according to which a matching matrix is constructed given a pair of input texts. The matching matrix, which has the property of hierarchical matching patterns, is then fed into a Convolutional Neural Network (CNN) to compute the matching score for the given user–item pair. Our experiments on the small data categories of Amazon review dataset show that our proposed model gains from 1.76% to 21.72% relative improvement compared to DeepCoNN model (Zheng et al., 2017), and from 0.83% to 3.15% relative improvement compared to TransNets model (Catherine and Cohen, 2017). Also, on two large categories, namely `AZ-CSJ` and `AZ-Mov`, our model achieves relative improvements of 8.08% and 7.56% compared to the DeepCoNN model, and relative improvements of 1.74% and 0.86% compared to the TransNets model, respectively.

1. Introduction

Today, with the rapid growth of the Internet and its availability, online platforms have been utilized much more than before, that as a result, has motivated providers to make their best efforts to increase customers satisfaction which is an important aspect of any successful business. In this light, recommender systems have been developed to prevent users from being overwhelmed by a large number of available items. To this aim, the recommender systems provide a personalized list of items based on users' interest. Various online platforms, including online shops, movie and song websites, hotel bookings, digital libraries, and news websites benefit from this facility to enhance users' experience of their service.

Recommender systems autonomously gather information on the preference of users for a set of items (e.g., hotels, books, movies, songs, etc.) upon which they proactively predict the preference each user would give to an item. Thus, the problem that recommender systems seek to solve could be viewed as a rating prediction problem. Today, with the prevalence of online services and due to the abundance of choice in quite all such services, recommender systems play an increasingly significant role.

As deep learning approaches have recently obtained considerably high performance across many different machine learning applications, recent research studies on recommender systems have also focused on using deep neural networks which have shown impressive results. In this approach, neural networks are mostly used to construct latent representations of users and items using the content associated with them. Content of a user or an item could include any property of each; e.g., demographic information and product preferences are potential content for users, whereas the content linked to items could include their price or any other attribute with respect to their application in general. To make rating prediction, latent representations are then used as input to a regression or collaborative filtering model. However, unlike the content information that describes only the user or only the item, review text could be considered a valuable common context information, featuring joint user–item interaction. Most neural recommender models such as those proposed by Bansal et al. [1], Elkahky et al. [2], Kim et al. [3], Li et al.

* Corresponding author.
E-mail address: momtazi@aut.ac.ir (S. Momtazi).

https://doi.org/10.1016/j.asoc.2020.106985
1568-4946/

[4] and Wang et al. [5] have focused on the content information of either users or items. On the other hand, only a limited number of neural recommender models such as proposed models by Almahairi et al. [6], Seo et al. [7], Zheng et al. [8], Catherine and Cohen [9], Li et al. [10] and Chen et al. [11] use user-provided reviews. These review-based models have shown that utilizing reviews could considerably improve the performance of recommendation compared to traditional approaches such as collaborative filtering techniques. Using the valuable information in reviews employed in such models have particularly proved to alleviate the well-known rating sparsity.

Most review-based neural recommender models work upon text matching approaches; i.e., they work similar to neural information retrieval models with the text corresponding to the user as the query and the text corresponding to the item as the document. The task of text matching is the central part of many natural language processing applications, such as question answering [12] and paraphrase identification [13]. In general, neural text matching models can be divided into two families: representation-based models and interaction-based models, which will be discussed in Section 3. The state-of-the-art neural recommender models which are based on text matching [7–9], use the representation-based learning approach. Although interaction-based models have been successfully used for different applications such as question answering, paraphrase identification, and document retrieval, none of the previous proposed neural recommender systems have used this approach. As reported in the literature, interaction-based approach achieved better performance compared to the representation-based approach, due to capturing more informative aspects of relevance between a pair of texts, which is a user document or an item document in our case [14].

In this paper, we propose to model the problem of rating prediction as a text matching problem using an interaction-based approach. The key contribution of our model is that it captures important matching patterns between user text and item text by a learning method on the basis of local pairwise interaction between every term of the user review text and every term of the item review text. Furthermore, the proposed model uses a regression layer on the top of the last representation layer of the joint user–item interaction to estimate their corresponding rating. To the best of our knowledge, this is the first research to use an interaction-based text matching approach for recommendation. In this formulation, each user is represented by all reviews she/he has written for different items, denoted as user document, and, similarly, each item is represented by all reviews written for it by different users, denoted as item document. Given the representations of all users and all items, our goal is to find the matching score for each pair of user–item for which we intend to determine whether to recommend the item to the user. Based on this idea, we believe that the matching score would be a good representative of the similarity between user and item which is essentially the main guideline for recommendation. Inspired by MatchPyramid [15], we employed a CNN architecture fed by the matching matrix of corresponding reviews for a pair of user–item. Our model, MPRS, achieves a better performance with respect to rating prediction compared to the state-of-the-art deep recommendation systems which benefit from user reviews too.

The rest of this paper is organized as follows: Section 2 describes related works. In Section 3, we note the difference between representation-based and interaction-based text matching approaches. The proposed model and architecture are introduced in detail in Section 4. The experiments and results are discussed in Section 5, and we conclude the paper in Section 6.

2. Related works

In this section, we provide an overview of the literature that benefits from user reviews for recommendation. We classify this line of research into two categories: non-neural models and neural models.

2.1. Non-neural models

The *Hidden Factors as Topics* (HFT) model by McAuley and Leskovec [16] works by regularizing the latent user and item parameters obtained from ratings with hidden topics in reviews. To this end, LDA topic modeling on reviews is combined with a matrix factorization model to be used as an objective function. A modified version of HFT is the *TopicMF* model by Bao et al. [17], where latent user and item factors learned using matrix factorization are jointly optimized with the topic modeling of their joint review. To this aim, they utilized the non-negative matrix factorization technique. Ling et al. [18] proposed the *Rating Meets Reviews* (RMR) model which extends the HFT model. In their proposed RMR model ratings are sampled from a Gaussian mixture.

2.2. Neural models

Almahairi et al. [6] used matrix factorization to learn the latent factors of users and items. This model benefits from review texts to overcome the data sparsity problem in matrix factorization technique. Following the multitask learning framework by Caruana [19], the model jointly predicts rating given by the user u to the item i and models the review written by the user u for the item i. The model consists of two components: (1) rating prediction, and (2) review modeling which shares some of the parameters from the former component. Almahairi et al. [6] then proposed two representations for modeling the likelihood of the review texts, namely bag-of-words and LSTM embeddings. Shoja and Tabrizi [20] proposed to first extract product attributes from user reviews using the Latent Dirichlet Allocation (LDA). With the extracted attributes, they constructed a users-attributes matrix. Then, to overcome the sparsity of the resulting users-attributes matrix, they used a neural network autoencoder that transforms the sparse matrix into a dense users-latent attributes matrix. Given the resulting dense matrix, matrix factorization is then employed to predict ratings. Ghasemi and Momtazi [21] proposed a model to extend collaborative filtering, such that user similarities are calculated based on reviews in addition to ratings. They used different neural text representations to calculate the similarity between user reviews.

Seo et al. [7] proposed attention-based CNNs to extract latent representations of users and items according to which the model makes rating prediction. In another research by Wang et al. [22], a hierarchical attention model is proposed and compared with general attention mechanism on user reviews. Their proposed hierarchical attention model aims at fusing latent factor models and capturing important words and informative reviews.

One of the recent neural models which efficiently predicts rating is the *Deep Cooperative Neural Networks* (DeepCoNN) model [8]. This model consists of two deep neural networks to obtain the latent representations of users and items from their corresponding reviews. User and item representations are then input to a layer to estimate their corresponding rating. A more recent study which successfully outperformed DeepCoNN on rating prediction is the *Transformational Neural Networks* (TransNets) model by Catherine and Cohen [9]. This model is an extension to DeepCoNN model by adding a latent layer to obtain an approximate representation of the review corresponding to

the user–item pair in the input. While training, this layer is regularized to be similar to the latent representation of the actual review of the target user–item pair which is computed through training a sentiment analysis network. The main idea is that the joint review of a user–item pair gives an insight into the user's experience with the item. At test time, the joint review is not given; therefore, an approximation of the user–item joint review could improve rating prediction.

In the proposed model by Liu et al. [23], a hybrid neural recommendation model is used which includes three major components: (1) a module that learns rating-based representation of users and items from their rating patterns, (2) a module that learns review-based representation of users and items from their respective text reviews, and (3) a rating prediction module that recommends based on both rating- and review-based representations of users and items. Furthermore, they proposed a review-level attention mechanism, that given the rating-based representation as input, selects useful reviews, in terms of informativeness, to be used for obtaining review-based representations in the second aforementioned module. Li et al. [10] proposed a neural recommendation system, named *Neural Rating and Tips Generation* (NRT), which jointly predicts ratings and generates abstractive tips using a multi-task learning approach. Chen et al. [11] introduced a neural attention mechanism to simultaneously learn the usefulness of each review and predict ratings. In this setting, highly-useful reviews are obtained to provide review-level explanations which help users to make better and faster decisions.

Wang et al. [24] proposed a recommendation method using convolutional matrix factorization. To model user reviews, they exploited a CNN model to learn users' posted contents. In addition to user reviews and item reviews, they benefit from users' information in social networks.

The mentioned models in the literature have built their architectures on the top of representation-based text matching; i.e., a general representation is built for each text and the two representations are then compared to find the relevance of a user to an item. Although this approach can capture the overall meaning of a text, considering the nature of reviews which are normally long and include discussions on various aspects of an item, the approach cannot capture local similarity of texts. On the contrary, the current research is a pioneer in utilizing an interaction-based text matching model for neural recommendation to benefit from more local text matching signals and better focus on different aspects of items that are mentioned in user reviews. In the next section, we explain representation-based and interaction-based text matching approaches in more detail.

3. Representation-based vs. interaction-based text matching

The task of text matching essentially involves computing the relevance between a given pair of texts. In ad-hoc information retrieval, the pair of texts consist of a query and a document. Representation-based text matching models work by learning meaningful representations for each text separately through applying several hidden layers on the text terms. The two representations are then fed into a similarity computing layer to estimate their relevance.

On the other hand, interaction-based text matching models capture the local interactions between two given texts by first calculating the similarity between each pair of words, and then learning the underlying higher level interaction patterns through several layers.

According to Nie et al. [14], in general, the interaction-based approach performs better than the representation-based approach in deep neural models for information retrieval. Since with representation-based models, it is very hard to learn global, informative representation of a user document or an item document which is usually very long, it is not surprising that representing every aspect of a long review document with a single vector is almost impossible. On the other hand, interaction-based models determine local matching signals between user document and item document by considering the interaction between every term of user document with every term of item document. It is important to note that traditional information retrieval models essentially work on the basis of similar local matching signals.

4. Model architecture

The main idea of the proposed model comes from modeling user–item similarity as text matching, since each user or item could be represented by the text of its corresponding reviews. We perform the text matching task by building a matching/interaction matrix as the joint representation of the user–item pair. To compute the matching score between a user text and an item text, we then use an interaction-based matching model on the top of the matching matrix. In this model, we represent the input of text matching as a matching matrix with entries standing as the similarity between words.

We consider each data entry as a tuple (j, k, r_{jk}, rev_{jk}) which denotes a review written by user j for item k with rating r_{jk} and the review text rev_{jk}. To construct the matching matrix, first all the reviews written by user j are concatenated together into a single document, denoted as $d^j_{1:n}$, consisting of n words. The same process is applied to get each item's corresponding document, denoted as $d^k_{1:m}$ for the item k, where m is the length of the document.

The input of our proposed model for the pair of user j and item k is a matching matrix, $M^{j,k}$, with each element $m^{j,k}_{p,q}$ as the similarity between the words w_p and v_q denoting the pth and qth words in $d^j_{1:n}$ and $d^k_{1:m}$ respectively. The similarity between two words can be computed by calculating the cosine similarity of their word embeddings to capture the semantic matching between words. Cosine similarity is calculated as follows:

$$m^{j,k}_{p,q} = \frac{\alpha^{j\top}_p \cdot \beta^k_q}{\|\alpha^j_p\| \, \|\beta^k_q\|} \tag{1}$$

where α^j_p and β^k_q are the word embeddings corresponding to w_p and v_q respectively. The matching matrix constructed in this way represents the joint representation of the user–item pair.

The architecture of our proposed model for rating prediction is shown in Fig. 1. In the first layer, the matching matrix for the given user–item pair is constructed, as it is described above. The matrix, which captures the joint semantic information in the review texts, is then fed into a CNN architecture followed by a regression network to predict the matching score for the corresponding user–item pair.

The CNN part consists of common layers of CNN-based models, including convolution layer, batch normalization layer, and max-pooling layer to learn the underlying interaction patterns. At the end, a fully-connected layer followed by a regression layer is added to the top of the last max-pooling layer to make a prediction on the rating $\hat{r}_{jk}$ that the user j would give to the item k.

The CNN models are used to extract effective features in a high-dimensional feature space which has disturbance and distortion. When the effective feature space is obtained, we can use fully-connected layers to perform the desired task of classification or regression. The key operations to extract effective features, or what is called dimension reduction, are filtering and

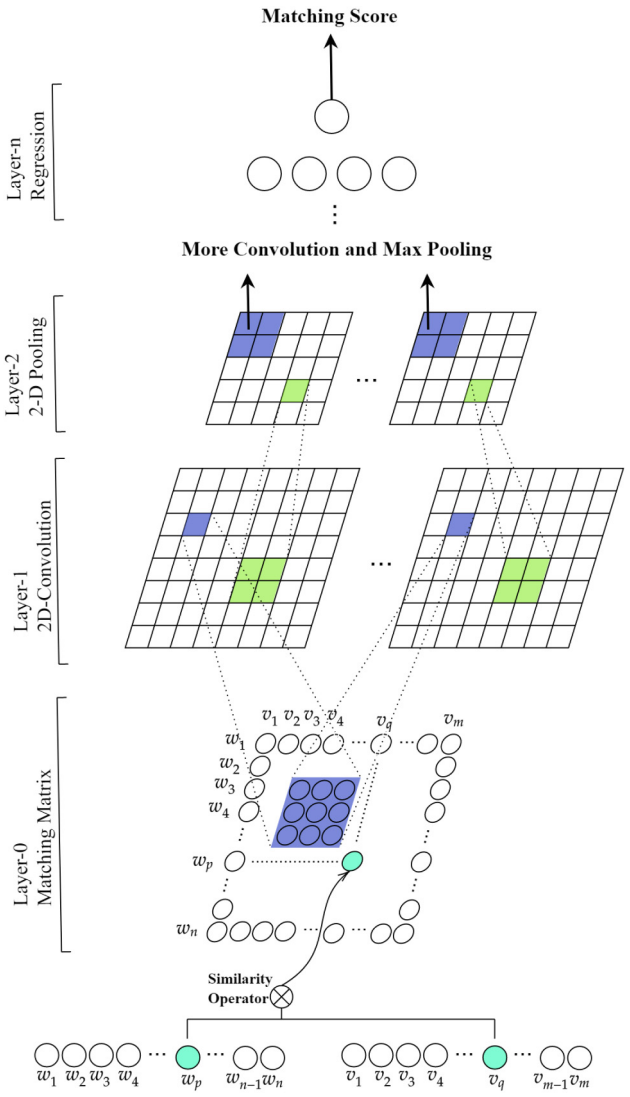

Fig. 1. The architecture of the proposed MPRS model.

scaling which are performed by convolution and pooling layers respectively.

Each convolution layer uses filter $K_i \in \mathbb{R}^{c\times t}$ to produce the feature map z_i which is defined as:

$$z_i = f(M^{j,k} * K_i + b_i) \tag{2}$$

where $M^{j,k} \in \mathbb{R}^{n\times m}$ is the input matching matrix, the symbol $*$ is the convolution operator, b_i is a bias term and f is an activation function. $z_i \in \mathbb{R}^{(n-c+1)\times(m-t+1)}$, if the stride is 1. For our model, we use the Rectified Linear Unit (ReLU) as the activation function which is defined as:

$$f(x) = \max(0, x) \tag{3}$$

It should be added that we perform a convolution operation regarding each kernel K_i in the convolution layer. Each convolution layer removes infrequent sub-patterns (disturbances) and extracts frequent sub-patterns.

We then apply a max-pooling operation over each feature map obtained from the convolution layer, and take the maximum value in each pooling window as one of the features for the corresponding kernel. In fact, each pooling layer reduces the dimension of the feature maps by sub-sampling. After each pooling operation, the extracted features are more invariant to distortion and scale.

By repeating the convolution and pooling layers, we allow filtering disturbances and concurrently extracting each feature with a certain scale. The result from the final max-pooling layer is passed to a fully connected layer, followed by a regression layer which is a single neuron layer with a linear activation function. Accordingly, the matching score $\hat{r}_{jk}$ is calculated as:

$$\hat{r}_{jk} = W \times O + g \tag{4}$$

where O is the result from the fully-connected layer, W contains the regression layer's weights and g is the bias.

In essence, the interaction structures in the process of text matching are compositional hierarchies of signals which might appear in the matching matrix, i.e. when matching two texts, word-level matching signals together form the phrase-level signals, and phrase-level signals assemble into sentence-level signals. Thus, the hierarchical convolutions performed in our model would capture the matching patterns between two texts.

The approach is designed based on a deep architecture for text matching, MatchPyramid [15]. The original architecture was proposed to predict if two texts are similar or not. In our model, however, instead of classifying two texts as relevant/similar or non-relevant/dissimilar, we aim at estimating the relevance degree of two texts. Therefore, our architecture is adapted to make regression prediction for the user–item similarity instead of classification.

Table 1
Dataset statistics separated by category.

Dataset	Category	#Users	#Items	#Ratings & Reviews
AZ-CSJ	Clothing, Shoes and Jewelry	3,116,944	1,135,948	5,748,920
AZ-Mov	Movies and TV	2,088,428	200,915	4,607,047
AZ-OP	Office products	4905	2420	53,258
AZ-IV	Amazon instant video	5130	1685	37,126
AZ-Auto	Automotive	2928	1835	20,473
AZ-PLG	Patio, Lawn and Garden	1686	962	13,272
AZ-MI	Music instruments	1429	900	10,261

5. Experiments

In order to evaluate the effectiveness of our proposed model, we performed different experiments. In this section, the dataset, the setup of the experiments as well as their results are presented.

5.1. Dataset

We evaluate the performance of our proposed model by using the most recent release of Amazon dataset[1] [25,26] which includes reviews and ratings given by users for products purchased on `amazon.com`. The dataset contains Amazon product reviews and metadata from May 1996 to July 2014, separated by category.

The main advantage of Amazon data is that it is not a single dataset, but a collection of datasets; i.e., each category of Amazon is considered as a separate dataset. To have a comprehensive analysis, we performed our experiments on 7 different categories with different characteristics. We included both large and small datasets with different distributions of users and reviews. In the following, we present the results of experiments on 7 different categories of this dataset. The statistics of these categories are given in Table 1.

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

Table 2
Review statistics separated by category.

Dataset	User review average length	Item review average length
AZ-CSJ	78.77	214.4
AZ-Mov	197.87	2047.3
AZ-OP	1298.4	2630.65
AZ-IV	545.69	1659.32
AZ-Auto	486.69	775.99
AZ-PLG	1015.79	1779.53
AZ-MI	526.52	835.4

The categories presented in the table are from both large and small Amazon datasets. As can be seen, the average number of review and rating pairs provided by each user on the large datasets is less than three, which shows that these categories are extremely sparse. This issue may considerably deteriorate the performance of recommender systems. More statistics about the length of reviews in these datasets are presented in Table 2.

5.2. Evaluation metric

In our experiments, we employed the Mean Squared Error (MSE) to evaluate the performance of our proposed model, as it is the sole metric that has been used for evaluation in most of the related works. Let N be the total number of datapoints in the test set. MSE can be computed as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(r_i - \hat{r}_i)^2 \tag{5}$$

where r_i is the ith observed value and $\hat{r}_i$ is the ith predicted value.

5.3. Setup of experiments

For each category, we divided the dataset into three sets of training, validation and test which are 80%, 10% and 10% of the whole dataset, respectively. Given training, validation and test sets, the document associated with each user or item is then constructed. Let rev_{jk} denote the review written by user j for item k. Let d^j and d^k be the documents to be constructed corresponding to $user_j$ and $item_k$ respectively. For all $(user_j, item_k)$ pairs in the test set, rev_{jk} is omitted from both d^j and d^k, since at test time, as the simulation of a real world situation, the joint review of a user–item pair is not obviously given. We build the train, validation and test sets accordingly for all our experiments for all the models including the baselines.

The parameters of our model are all chosen through a grid search on the validation sets. Accordingly, we set the number of convolution layers and the number of convolutional kernels for each layer to 7 and 32 respectively. Other hyper-parameters including the batch size and the dropout rate are set as 64 and 0.8 respectively. For the representation of the words, we used pre-trained 300-dimensional word embeddings [27] which are trained on part of Google News dataset[2] containing about 100 billion words.

5.4. Baselines

In order to evaluate the performance of our model, we selected DeepCoNN [8] and TransNets [9] to compare our results with. They are two of the most recent neural recommender models known as state-of-the-art models in this field. The performance of the DeepCoNN model reported in work by Zheng et al. [8] corresponds to an obsolete version of the dataset to which we had no access, therefore, we implemented the DeepCoNN model and tested the model on the latest version of the Amazon dataset. For the TransNets model and its variant, TransNet-Ext, we used the provided code by the authors that is available on `github`.[3]

Table 3
Performance of the proposed model compared to the state-of-the-art baselines using MSE.

Dataset	DeepCoNN	TransNets	TransNet-Ext	MPRS (relative improvement)
AZ-CSJ	1.5487	1.4487	1.4780	1.4235 (1.74%)**
AZ-Mov	1.3611	1.3599	1.2691	1.2582 (0.86%)*
AZ-OP	0.7566	0.8463	0.7495	0.7433 (0.83%)*
AZ-IV	1.1052	1.0564	1.0282	1.003 (2.45%)**
AZ-Auto	1.1758	0.9735	0.9425	0.9204 (2.34%)**
AZ-PLG	1.3136	1.0958	1.0852	1.0572 (2.58%)**
AZ-MI	1.0705	0.9185	0.9152	0.8864 (3.15%)**

*Marks statistical significant difference between the proposed model and the best result from the state-of-the-art baselines at $p < 0.05$ based on 2-tailed paired t-test.
**Shows highly statistical significance at $p < 0.01$ based on 2-tailed paired t-test.

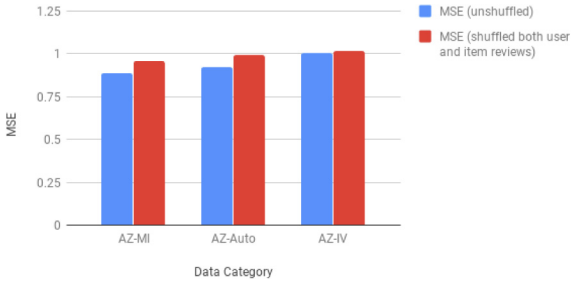

Fig. 2. The impact of shuffling reviews in user and item documents on MSE.

5.5. Results

The MSE values of MPRS, DeepCoNN, TransNets and TransNet-Ext models on 7 different categories of the Amazon dataset are given in Table 3.

As it is shown in the table, our proposed model predicts ratings better than the competitive baselines on all the categories. The numbers in the parentheses show the relative improvement compared to the best result in the baseline models. As can be seen, our model gains the maximum relative improvement (3.15%) on `AZ-MI` category which is the smallest data category in terms of the total number of reviews. It indicates that our model can achieve superior results even on a small number of instances by taking into account the interaction between user text and item text. Overall, the relative improvements of our model on two large datasets are 0.86% and 1.74%, and the relative improvement on small datasets is from 0.83% to 3.15% .

These results validate our hypothesis that capturing local text matching signals provide additional information for computing the similarity between user and item. Considering that users normally talk about different aspects of an item in their review, a single representation vector cannot accurately capture these aspects to find the relevance between users and items. The interaction-based approach, however, can match different parts of reviews and find the relation between users and items by capturing local interactions between review texts. The experimental results show that the proposed model can improve rating prediction due to its advanced architecture.

[2] https://code.google.com/archive/p/word2vec/.

[3] https://github.com/rosecatherinek/TransNets.

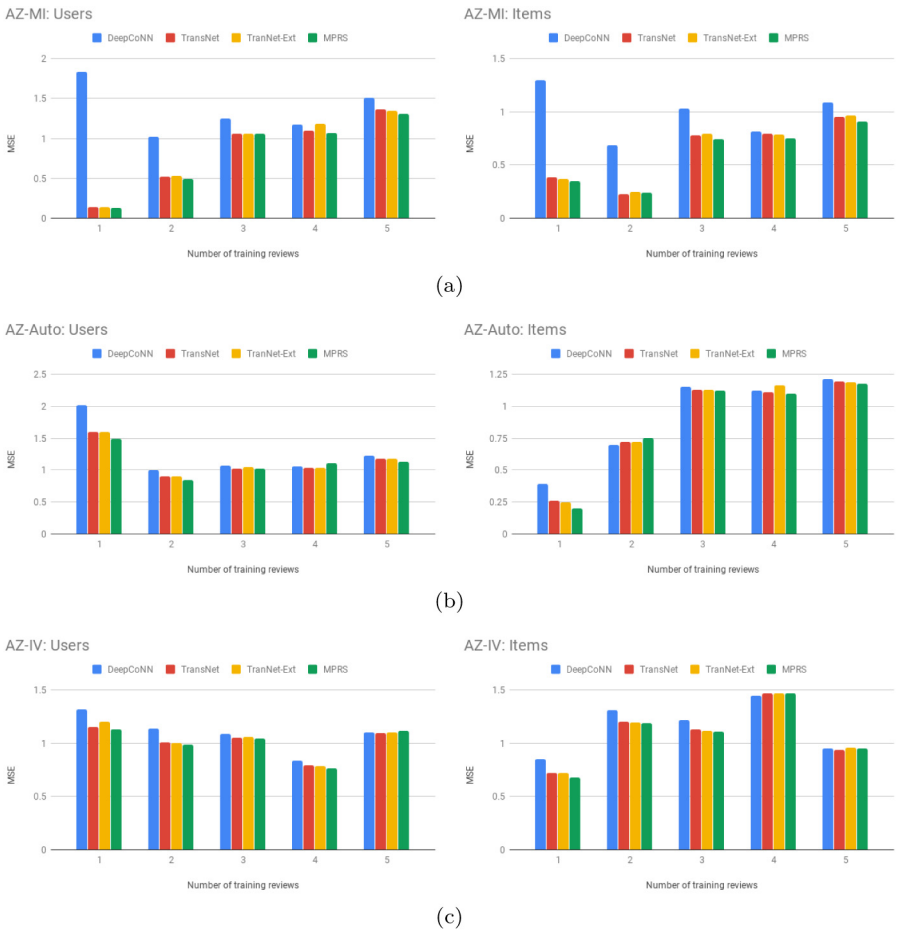

Fig. 3. MSE values of MPRS compared to DeepCoNN, TransNets and TransNet-Ext for users and items groups with different number of training reviews on three data categories.

In the next step of our experiments, we evaluate the robustness of our proposed model in case of changing the order of reviews in user/item documents. According to the structure of the problem and the way we represent users and items in this work, it is expected that if we change the order of reviews in the user document or item document, the model's prediction of rating does not undergo considerable changes. On the other hand, by changing the order of reviews in a document the matching matrix changes as well. As a result, the input to our model would be a different matrix and the following question is raised: "given our model trained with a specific order of reviews in each user/item document, if we change the order of reviews in user document and item document in test set, does the model's performance in rating prediction undergo considerable changes?".

Fig. 2 shows the MSE values of the MPRS model on three different categories of Amazon dataset. For each category, there are two MSE values reported, one corresponds to the MSE of MPRS on the test set with the same order reported in Table 3 and the other with a shuffled document of each user and item. As can be seen in Fig. 2, for all three categories the amount of changes in the MSE value as a result of changing the order of reviews in documents is insignificant.

In the next part of our experiments, we compare the performance of our proposed model with the baseline models in case of data sparsity. Some users might rate only a limited number of items. The data sparsity problem arises when there is not enough rating data available for some users and items, therefore recommender systems are not able to learn preferences regarding such users and items. Cold start problem, specifically refers to the difficulty of recommendation to new users and items due to lack of data. The data sparsity problem is one of the main challenges in designing recommender systems. Recent studies have shown that using reviews can help to considerably resolve the data sparsity problem, especially for users or items with few ratings [8,16]. The diagrams in Fig. 3 depict the MSE values of MPRS, DeepCoNN , TransNets and TransNet-Ext models on three categories of the Amazon dataset. To test the models' performances in this condition, for each dataset category, users and items are grouped based on the number of reviews in the training set. In this experiment, the maximum number of training reviews is set to five. For each data category, MSE values of the models are plotted for both users and items groups. As it is shown in Fig. 3, our proposed model performs better than the baseline models on all three data categories for almost any number of training reviews (1–5). Especially, when having only one review available, which is the worst case of data sparsity in this formulation, our model significantly outperforms the baselines.

6. Conclusion

In this paper, we propose a novel neural recommender system which recommends items by leveraging user reviews. We represent each user and item with their corresponding reviews and view the problem of recommendation as a text matching problem. We used an interaction-based model to address the text matching problem. To this aim, a CNN architecture followed by a regression network was employed to predict user–item pair matching score as a representative of user's rating for item.

Experimental results on various data categories of Amazon show that our proposed model outperforms the state-of-the-art neural recommender systems which are based on reviews.

There are multiple possible directions to extend the model proposed in this paper. One way to extend the model is to use an attention-based CNN to locate the attention to most representative parts of the user–item matching matrix which would probably improve the performance of the base model. Another approach to further improve the performance of the model is to use a target network as well as the main CNN which should be followed by a Transform layer as inspired by the TransNets model [9]. The Transform layer transforms user–item joint feature maps obtained from the CNN into an approximation of their joint review which would give an insight into the user's experience with the item during training as well as evaluation. The target network as used in the TransNets model, provides the latent representation of target joint review to train the main CNN and the Transform layer.

CRediT authorship contribution statement

Parisa Abolfath Beygi Dezfouli: Conception and design of study, Acquisition of data, Implementation of the proposed model, Analysis and/or interpretation of data, Drafting the manuscript. **Saeedeh Momtazi:** Conception and design of study, Acquisition of data, Supervision, Editing the manuscript. **Mehdi Dehghan:** Supervision, Editing the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] T. Bansal, D. Belanger, A. Mccallum, Ask the GRU: Multi-task learning for deep text recommendations, in: Proceedings of the 10th ACM Conference on Recommender Systems - RecSys 16, 2016.

[2] A.M. Elkahky, Y. Song, X. He, A multi-view deep learning approach for cross domain user modeling in recommendation systems, in: Proceedings of the 24th International Conference on World Wide Web - WWW 15, 2015.

[3] D. Kim, C. Park, J. Oh, S. Lee, H. Yu, Convolutional matrix factorization for document context-aware recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16, 2016.

[4] S. Li, J. Kawale, Y. Fu, Deep collaborative filtering via marginalized denoising auto-encoder, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM 15, 2015, pp. 811–820.

[5] H. Wang, N. Wang, D.Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 15, 2015, pp. 1235–1244.

[6] A. Almahairi, K. Kastner, K. Cho, A. Courville, Learning distributed representations from reviews for collaborative filtering, in: Proceedings of the 9th ACM Conference on Recommender Systems - RecSys 15, 2015, pp. 147–154.

[7] S. Seo, J. Huang, H. Yang, Y. Liu, Representation learning of users and items for review rating prediction using attention-based convolutional neural network, in: Proceedings of the 3rd International Workshop on Machine Learning Methods for Recommender Systems - MLRec, 2017.

[8] L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: Proceedings of the 10th ACM International Conference on Web Search and Data Mining - WSDM '17, 2017, pp. 425–434.

[9] R. Catherine, W. Cohen, TransNets: Learning to transform for recommendation, in: Proceedings of the 11th ACM Conference on Recommender Systems - RecSys '17, 2017, pp. 288–296.

[10] P. Li, Z. Wang, Z. Ren, L. Bing, W. Lam, Neural rating regression with abstractive tips generation for recommendation, in: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, 2017, pp. 345–354.

[11] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1583–1592.

[12] X. Xue, J. Jeon, W.B. Croft, Retrieval models for question and answer archives, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2008, pp. 475–482.

[13] R. Socher, E.H. Huang, J. Pennin, C.D. Manning, A.Y. Ng, Dynamic pooling and unfolding recursive autoencoders for paraphrase detection, in: Advances in Neural Information Processing Systems, 2011, pp. 801–809.

[14] Y. Nie, Y. Li, J.Y. Nie, Empirical study of multi-level convolution models for ir based on representations and interactions, in: Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, 2018, pp. 59–66.

[15] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, X. Cheng, Text matching as image recognition, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence - AAAI'16, 2016, pp. 2793–2799.

[16] J. McAuley, J. Leskovec, Hidden factors and hidden topics: Understanding rating dimensions with review text, in: Proceedings of the 7th ACM conference on Recommender systems - RecSys '13, 2013, pp. 165–172.

[17] Y. Bao, H. Fang, J. Zhang, TopicMF: Simultaneously exploiting ratings and reviews for recommendation, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence - AAAI'14, 2014, pp. 2–8.

[18] G. Ling, M.R. Lyu, I. King, Ratings meet reviews, a combined approach to recommend, in: Proceedings of the 8th ACM Conferenceon Recommender Systems - RecSys '14, 2014, pp. 105–112.

[19] R. Caruana, Multitask learning, Mach. Learn. 28 (1997) 41–75.

[20] B.M. Shoja, N. Tabrizi, Customer reviews analysis with deep neural networks for e-commerce recommender systems, IEEE Access 7 (2019) 119121–119130.

[21] N. Ghasemi, S. Momtazi, Neural text similarity of user reviews for improving collaborative filtering recommender systems, Electron. Commer. Res. Appl. (2020) 101019.

[22] X. Wang, H. Liu, P. Wang, F. Wu, H. Xu, W. Wang, X. Xie, Neural review rating prediction with hierarchical attentions and latent factors, in: G. Li, J. Yang, J. Gama, J. Natwichai, Y. Tong (Eds.), Database Systems for Advanced Applications - 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22–25, 2019, Proceedings, Part III, and DASFAA 2019 International Workshops: BDMS, BDQM, and GDMA, Springer, Chiang Mai, Thailand, 2019, pp. 363–367, Proceedings.

[23] H. Liu, Y. Wang, Q. Peng, F. Wu, L. Gan, L. Pan, P. Jiao, Hybrid neural recommendation with joint deep representation learning of ratings and reviews, Neurocomputing 374 (2020) 77–85.

[24] X. Wang, X. Yang, L. Guo, Y. Han, F. Liu, B. Gao, Exploiting social review-enhanced convolutional matrix factorization for social recommendation, IEEE Access 7 (2019) 82826–82837.

[25] J. McAuley, R. Pandey, J. Leskovec, Inferring networks of substitutable and complementary products, in: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15, 2015, pp. 785–794.

[26] J. McAuley, C. Targett, Q. Shi, A. van den Hengel, Image-based recommendations on styles and substitutes, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15, 2015, pp. 43–52.

[27] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.