J3R: Joint Multi-task Learning of Ratings and Review Summaries for Explainable Recommendation

Avinesh P.V.S.[1]✉, Yongli Ren[2], Christian M. Meyer[1], Jeffrey Chan[2], Zhifeng Bao[2], and Mark Sanderson[2]

[1] Research Training Group AIPHES and UKP Lab
Computer Science Department, Technische Universität Darmstadt
www.aiphes.tu-darmstadt.de,www.ukp.tu-darmstadt.de
{avinesh, meyer}@ukp.informatik.tu-darmstadt.de
[2] Computer Science, School of Science, RMIT University, Australia
{yongli.ren, jeffrey.chan, zhifeng.bao, mark.sanderson}@rmit.edu.au

Abstract. We learn user preferences from ratings and reviews by using multi-task learning (MTL) of rating prediction and summarization of item reviews. Reviews of an item tend to describe detailed user preferences (e.g., the cast, genre, or screenplay of a movie). A summary of such a review or a rating describes an overall user experience of the item. Our objective is to learn latent vectors which are shared across rating prediction and review summary generation. Additionally, the learned latent vectors and the generated summary act as explanations for the recommendation. Our MTL-based approach J3R uses a multi-layer perceptron for rating prediction, combined with pointer-generator networks with attention mechanism for the summarization component. We provide empirical evidence for joint learning of rating prediction and summary generation being beneficial for recommendation by conducting experiments on the Yelp dataset and six domains of the Amazon 5-core dataset. Additionally, we provide two ways of explanations visualizing (a) the user vectors on different topics of a domain, computed from our J3R approach and (b) a ten-word review summary of a review and the attention highlights generated on the review based on the user–item vectors.

Keywords: Personalized Recommendation · Summarization · Natural Language Processing · Explainable AI

1 Introduction

Product recommender systems have increasingly gained attention in the Information Retrieval and Natural Language Processing communities, both in academia and industry. Most existing recommendation methods are based on collaborative filtering [21, 10, 9], which primarily learn users' and items' latent factors from ratings. Such an approach fails to capture valuable information from actual user experiences, which can be recorded in the form of reviews. This user-generated

User	Aspect Words
T.	Cast: fit, work Genre: action
hairbear2012	Genre: Spy, action Screenplay: storytelling
scooby	Screenplay: car chase Cost: bucks
Alex	Screenplay: chase scenes, human interaction

Fig. 1. Example ratings, reviews and their summaries for Jason Bourne (2016) on Amazon Movies. Reviews describe detailed personalized opinions and interests of the user w.r.t. the item. The table on the right-hand side shows extracted aspect words from the reviews modeling the users' preferences.

content is an increasingly important source, useful for both businesses as well as the end user. In this paper, we propose J3R, a novel multi-task learning setup for explainable recommendation based on ratings *and* reviews, which we motivate below.

User and item profiles for recommendation. Although recommender systems based on reviews have been previously proposed, [18, 16, 4, 27], they yet do not fully exploit the potential of learning to recommend jointly from both reviews and ratings. Figure 1 shows four reviews on the Jason Bourne (2016) movie, which illustrate the connection between reviews and ratings: Each review consists of a brief summary (e.g., "Better with age" in T.'s review) and the actual review text in addition to the rating (i.e., 1–5 stars). The users focus on multiple different aspects in their reviews. For example, user T. likes Matt Damon's looks, fitness, and the action in the movie. In contrast, Alex and scooby have differing opinions on the use of car chases in the screenplay. The example shown is a typical real-world use case where different users have different interests and opinions about certain aspects of the same item. We aim at exploiting this information from reviews for building user and item profiles. Additionally, we leverage recent advances in deep neural networks to exploit the commonality between the rating and the review summary in a multi-task learning (MTL) approach where rating prediction is the main task and review summary generation is the auxiliary task.

Explainable recommendation. In a recent review by [7] on European Union regulations on algorithmic decision-making, the authors explain how the Article 22 of the European Union's new General Data Protection Regulation on automated individual decision-making and profiling potentially prohibits a wide range of algorithms currently in use, including recommendation systems, computational advertising, etc. The law effectively states "the right to explanation", where a

user could ask for explanations on the decisions made about them by the algorithm. This regulation is only one recent development to strongly encourage the machine-learning-based communities to design algorithms in view of enabling explanations.

Although the primary goal of a recommender system is to produce excellent recommendations, it is a clear advantage if a system provides explanations for its users. Explanations serve multiple purposes, such as building trust, creating transparency, improving efficiency by quicker decision-making, and increasing user satisfaction [24]. There has been a recent surge in methods focusing on explainable recommendation systems [26, 3, 12, 4]. Previous approaches use explicit topics from reviews with users' opinions [26], knowledge graphs [3], tip generation [12] and review ranking [4] for explanations.

In our research, we propose a novel approach to combine explicit topic vectors from reviews with generated review summaries as a way to explain a predicted rating. The final explanations of our J3R system are thus of two types: (a) a histogram of user preferences on different topics of a domain, computed from the updated user vectors learned by our MTL approach and (b) a ten-word review summary of a review and the attention highlights on the review based on the weights learned from the user–item vectors. For the Jason Bourne example from Figure 1, a user vector for user T. should capture T.'s interest in the cast and the genre based on the user's past reviews. In addition to these histograms, based on the preferences from scooby's vector, the words in Alex's review would be highlighted according to their importance with respect to scooby's profile and the review would be automatically summarized.

Contributions. In this work, (1) we propose a novel combination of multi-layer perceptron and pointer-generator networks for jointly learning the shared users' and items' latent factors using an MTL approach for predicting user ratings and review summary generation – two related tasks that can mutually benefit from each other. (2) Our approach provides a way to explain the predicted ratings with the help of generated summaries and topic histograms, which further enhances the idea of evidence-based recommendation and decision-making. To encourage the research community and to allow for replicating our results, we publish our code as open-source software.[3]

2 Related Work

Previous works address recommendation systems employing (1) content-based filtering, (2) joint models of rating prediction and summary generation, and (3) explainable recommendation.

Content-based filtering. Collaborative filtering methods have seen successful for a long time in recommendation systems [21, 10, 9]. Methods like probabilistic matrix factorization (PMF) [21], non-negative matrix factorization (NMF) [10],

[3] `https://github.com/AIPHES/ecml-pkdd-2019-J3R-explainable-recommender`

singular value decomposition (SVD), and SVD++[9] have been successfully applied by representing users and items in a shared, low-dimensional space. Vectors in this space represent latent factors of users and items. Using the dot product of two vectors, one can predict a user's rating for a given item. A drawback of these approaches is that the system performance degrades when the rating matrix is sparse, which is often the case for newly developed systems or small communities. Furthermore, the vectors of this latent space have no particular interpretation, which impedes providing an explanation for a recommendation that can be understood by human users.

This propelled researchers towards content-based filtering techniques for recommendation [18, 16, 1]. Content-based filtering methods typically learn user [16] and item profiles [1] from item descriptions or user reviews. They recommend an item to a user by matching the item's features with that of the user preferences. There are works which identify the importance of aspects for the users by integrating topic models to generate the users' and items' latent factors from review text [18]. Our proposed approach also employs topic models to initialize latent user and item vectors, but we further update them by jointly training for rating prediction and review summary generation.

Joint models for rating prediction and summary generation. Multi-task learning approaches have seen significant success in the area of machine learning and natural language processing [19]. The goal of these approaches is to learn two related tasks which can mutually benefit from each other. As rating prediction and review summary generation are two facets of the same user preference of an item, they can be optimized together by sharing the parameters across the model. Although review summary generation has been conducted independently of rating predictions [28], jointly modeling the rating prediction and the review summary generation has as yet only shown first promising results [12, 25]. In our work, we go beyond such models by employing pointer-generator neural models and an attention mechanism on user preferences which particularly benefit the auxiliary task of review summary generation.

Explainable recommendation. Although state-of-the-art methods produce generally good recommendations, they fail to explain the reasons for a particular recommendation. Explanations can serve as a way to understand the algorithms and the models learned. This has led to new research questions for explaining recommendation systems and their output [3, 12, 17, 11]. Some of the promising approaches include topic models as latent factors [17], knowledge graphs [3], and tip generation [12, 11]. [17] propose a joint model using reviews and ratings with a Hidden Markov Model and Latent Dirichlet Allocation (LDA). They provide explanations with the help of words from latent word clusters explaining the essential aspects of the user and item pairs. [26] propose explicit factor models for generating explanations by extracting phrases and sentiments from user-written reviews for the items. In our approach, we combine multiple types of explanations and we generate them by jointly learning from reviews and ratings.

The work by [12] first proposes a multi-task learning framework to predict ratings and generate abstractive review summaries, which they extended in [11] by proposing a personalized solution. A major difference between their task and ours is that we generate summaries from the reviews, whereas they generate summaries from user–item latent vectors and the review vocabulary. Thus, the summaries generated in their task tend to be overly general as discussed in their paper. On the contrary, in this paper, our goal is not only to generate summaries but also to use summarization as a method to explain the important content in the reviews based on the user preferences. We leverage recent machine learning advances in pointer-generator networks [22] and attention-based mechanisms [20] which supports the accurate generation of summaries by attending on latent user–item vectors, the users' ratings, and their reviews.

3 Approach

We divide our proposed approach into the three components shown in Figure 2: (1) First, we build *user and item models* to identify interpretable topic vectors of an item capturing different aspects of the item that users are interested in. (2) Then, we train a *rating prediction model* using these user and item models. (3) Finally, we generate review summaries to explain the recommendations of our system by jointly modeling *rating prediction* and *review summary generation*, using an MTL approach of multi-layer perceptron and pointer-generator networks that utilizes the user and item models. Our final method is called J3R ('**J**oint MTL of **R**atings and **R**eview Summaries for Explainable **R**ecommendation'). We introduce the three components in the following subsections.

3.1 User and Item Models Component

The goal of the first component is to build user and item profiles using the review content. To achieve this goal we first preprocess the data to identify all nouns and noun phrases from the reviews (e.g., 'display', 'battery for a phone') similar to [14]. We collect the nouns in a bag-of-words representation to generate a 1,000-dimensional tf-idf vector, capturing the most frequent nouns describing an item in the given domain. These fixed-size tf-idf vectors are used as input for the LDA [2] topic model to calculate topic vectors. LDA is a probabilistic topic model which aims at finding structures in an unlabeled text collection by identifying different topics based on the word usage. The probability distribution over high probability words gives us an understanding of the contents of the corpus. Thus, reviews grouped into different clusters using LDA can be viewed as random mixtures over latent vectors, where a distribution over the most frequent nouns represents each topic.

Let D be a corpus of M reviews $D_1, D_2, \ldots, D_M$, where each review $D_i = (w_1, w_2, \ldots, w_N)$ is a sequence of N words from a vocabulary $\mathcal{W}$ and k the number of topics. Using LDA, we represent each document D_i as a k-dimensional

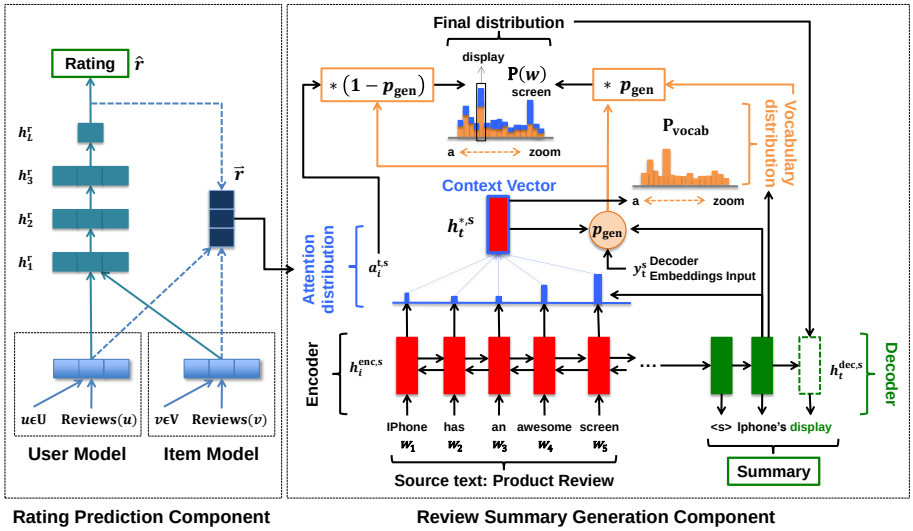


Fig. 2. Model architecture of our joint model for rating prediction and review summarization (J3R). The architecture is divided into three components: (1) user and item models, (2) rating prediction, (3) review summary generation.

topic distribution θ_d. Each topic vector, in turn, is an N-dimensional word distribution ϕ_k, which follows a Dirichlet prior β.

There are three steps to LDA: (1) it first draws a k-dimensional topic mixing distribution $\theta_d \sim Dir(\alpha)$ to generate a document d; (2) for each token w_{dn}, it draws a topic assignment z_{dn} from a multinomial distribution $Mult(\phi_{z_{dn}})$; and (3) finally, it draws a word $w_{dn} \in \mathcal{W}$ from $Mult(\phi_{z_{dn}})$ by selecting a topic z_{dn}. To infer these latent variables (ϕ and θ) and hyperparameters (α and β), we compute the probability of the observed corpus:

$$p(D|\alpha,\beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d)\, p(w_{dn}|z_{dn},\beta) \right) \mathrm{d}\theta_d \qquad (1)$$

We use all the reviews $Reviews_u$ written by a user u and all reviews $Reviews_v$ of an item v and turn them into N-dimensional tf-idf vectors. To generate topic vector profiles, we input these tf-idf vectors to the learned LDA topic model. The profiles learned using the user and item model are the initial latent vectors $u \in \mathcal{R}^k$ and $v \in \mathcal{R}^k$ for the rating prediction model discussed in the next section and are illustrated in Figure 2 as *User Model* and *Item Model*.

3.2 Rating Prediction Component

Our rating prediction component is illustrated on the left-hand side of Figure 2. It uses a traditional recommendation setup where the goal of the recommender is

to predict the rating of a given user and item pair. We use a regression function to predict a rating score $\hat{r}$ based on the latent vector representations u and v of the users and items. Typical matrix factorization (MF) approaches do a linear transformation of these vectors as described in Eq. 2, where b is the global bias.

$$\hat{r} = u^{\mathrm{T}} v + b \tag{2}$$

Although these linear transformations achieve state-of-the-art performance in recommendation systems, they cannot capture non-linear interactions between the users' and items' latent factors. Thus, we transfer knowledge from successful non-linear deep learning methods used in natural language processing for our task by concatenating the input vectors u and v as in Eq. 3:

$$h_1^r = relu(W_{h_1}^{\mathrm{r}}(u \oplus v) + b_{h_1}^{\mathrm{r}}) \tag{3}$$

where $W_{h_1}^{\mathrm{r}}$ is the weight matrix of the first hidden layer for the concatenated vector $u \oplus v$, u is the user's latent factors, and v is the item's latent factors. $b_{h_1}^{\mathrm{r}}$ is the bias term and $relu(x) = x^+ = \max(0, x)$ is the non-linear function. The superscript $^{\mathrm{r}}$ represents the parameters and variables for the rating prediction component of our model. To further add non-linearity, we add additional layers of non-linear transformations:

$$h_\ell^{\mathrm{r}} = relu(W_{h_\ell}^{\mathrm{r}} h_{\ell-1}^{\mathrm{r}} + b_{h_\ell}^{\mathrm{r}}) \tag{4}$$

where ℓ is the index of the hidden layer and $W_{h_\ell}^{\mathrm{r}}$ is the corresponding weight matrix. The number of hidden layers is a hyperparameter of our model.

Eq. 5 describes the output layer with the weight matrix $W_{h_L}^r$. We use a sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ to output a rating in the range [0,1], which we denormalize to the rating range (e.g., 1–5 stars) during the evaluation.

$$\hat{r} = \sigma(W_{h_L}^{\mathrm{r}} h_L^{\mathrm{r}} + b_{h_L}^{\mathrm{r}}) \tag{5}$$

To optimize the parameters and the latent factors u and v, we define the loss function:

$$\mathcal{L}^{\mathrm{r}} = \frac{1}{\mid \mathcal{X} \mid} \sum_{u \in \mathcal{U}, v \in \mathcal{V}} (\hat{r}_{u,v} - r_{u,v})^2 \tag{6}$$

where $\mathcal{X}$ is the training set, $\hat{r}_{u,v}$ is the predicted rating and $r_{u,v}$ is the gold-standard rating assigned by user $u \in \mathcal{U}$ to item $v \in \mathcal{V}$.

3.3 Review Summary Generation Component with Attention on User Preferences

The goal of J3R is to mutually benefit from the available ratings and reviews in two different tasks: (a) rating prediction and (b) review summary generation. Rating prediction precisely aims at predicting the score for a given user and item pair, whereas the review summary generation component summarizes the review content using a sequence-to-sequence model based on user preferences.

The user–item preferences (i.e., the user and item vectors) are shared with the rating prediction component, which are jointly learned using an MTL approach.

Our model is inspired by pointer-generator networks [22] to efficiently summarize the review, by using soft switching between copying words via pointing to the source text and generating words via a fixed vocabulary in a given context. The context in our generation setup consists of the user and item latent vectors $u \in \mathcal{U}$, $v \in \mathcal{V}$, the rating vector $\boldsymbol{r}$ (e.g., if the rating range is [1,5] then a rating vector for 3 stars is (0, 0, 1, 0, 0)), and the review D. The tokens of the review text $w_i \in D$ are provided as the input to the encoder one-by-one to produce a sequence of encoder hidden states $h_i^{\text{enc,s}}$. At each time step t, the decoder has the decoder states $h_t^{\text{dec,s}}$ which receives the word embeddings of the previous word as the input.

An important characteristic of our architecture is the attention distribution $a_i^{\text{t,s}}$ that we compute at each time step t with the encoder states $h_i^{\text{enc,s}}$, the decoder state $h_t^{\text{dec,s}}$, the user vector u, the item vector v, and the rating vector $\boldsymbol{r}$ as shown in Eq. 7–9. It can be viewed as a probability distribution over the source words, user preferences, item factors and rating, which tells the decoder which word to generate.

$$e_i^{\text{t,s}} = q^T \tanh(W_h^{\text{enc,s}} h_i^{\text{enc,s}} + W_h^{\text{dec,s}} h_t^{\text{dec,s}} + W_r^{\text{s}}(u \oplus v \oplus \boldsymbol{r}) + b_{\text{att}}^{\text{s}}) \tag{7}$$

$$a_i^{\text{t,s}} = \frac{\exp(e_i^{\text{t,s}})}{\sum_{i'=1}^{N} \exp(e_{i'}^{\text{t,s}})} \tag{8}$$

where q, $W_h^{\text{enc,s}}$, $W_h^{\text{dec,s}}$, W_r^{s} and $b_{\text{att}}^{\text{s}}$ are learnable parameters and N is the number of words in the review text. The superscript $^{\text{s}}$ represents the parameters and variables for the review summary generation component of our model.

Using the attention distribution $a_i^{\text{t,s}}$, we compute the weighted sum of the encoder hidden states, also known as the context vector $h_t^{*,\text{s}}$ as shown in Eq. 9.

$$h_t^{*,s} = \sum_i a_i^{\text{t,s}} h_i^{\text{enc,s}} \tag{9}$$

To get the vocabulary distribution P_{vocab} at time step t, we concatenate the context vector with the decoder state $h_t^{\text{dec,s}}$ and pass it through two linear layers:

$$P_{\text{vocab}} = softmax(Q\,(Q' h_t^{\text{dec,s}} \oplus h_t^{*,\text{s}} + b'^{\text{s}}) + b^{\text{s}}) \tag{10}$$

where Q, Q', b^{s} and b'^{s} are learnable parameters.

To finally generate words, we use a pointer-generator network which decides whether to generate the word from the vocabulary P_{vocab} or copy one from the input sequence by sampling from the attention distribution $a^{\text{t,s}}$ as shown in Eq. 12. This is done by calculating an additional generation probability $p_{\text{gen}}^{\text{s}}$ for time step t, which is calculated from the context vector $h_t^{*,\text{s}}$, the decoder state $h_t^{\text{dec,s}}$,

and the current input to the decoder y_t^{s}:

$$p_{\mathrm{gen}} = \sigma(W_{h^*}^T h_t^{*,\mathrm{s}} + W_{h^{\mathrm{dec}}}^T h_t^{\mathrm{dec,s}} + W_y^T y_t^{\mathrm{s}} + b_{\mathrm{gen}}^{\mathrm{s}}) \tag{11}$$

$$P(w) = p_{\mathrm{gen}} P_{\mathrm{vocab}}(w) + (1 - p_{\mathrm{gen}}) \sum_{i=0}^{N} a_i^{\mathrm{t,s}} \tag{12}$$

where W_{h^*}, $W_{h^{\mathrm{dec}}}$, W_y, $b_{\mathrm{gen}}^{\mathrm{s}}$ are learnable parameters and N is the number of words in the source review. Pointer-generator networks are helpful for handling out-of-vocabulary (OOV) words: if w is an OOV word then $P_{\mathrm{vocab}} = 0$ and the word from the source review text is considered for generation and vice versa.

Finally, we define the loss function for the review summary generation component for the whole sequence as the normalized sum of the negative log likelihood of the target words w_t^*:

$$\mathcal{L}^{\mathrm{s}} = -\frac{1}{T} \sum_{t=0}^{T} \log P(w_t^*) \tag{13}$$

3.4 Multi-task Learning Setup

We use a multi-task learning setup to jointly optimize the rating prediction and the review summary generation components by using a joint loss function $\mathcal{L}^{\mathrm{j}}$:

$$\mathcal{L}^{\mathrm{j}} = \lambda_{\mathrm{r}} \mathcal{L}^{\mathrm{r}} + \lambda_{\mathrm{s}} \mathcal{L}^{\mathrm{s}} + \lambda_{\mathrm{o}} \left({||\mathcal{U}||_2}^2 + {||\mathcal{V}||_2}^2 + {||\Omega||_2}^2 \right) \tag{14}$$

where $\mathcal{L}^{\mathrm{r}}$ is the rating regression loss from Eq. 6 and $\mathcal{L}^{\mathrm{s}}$ is the review summary generation loss from Eq. 13. For regularization, we use l_2-norm on the set of neural network parameters Ω, the user latent factors $\mathcal{U}$ and the item latent factors $\mathcal{V}$. λ_{r}, λ_{s}, λ_{o} are hyperparameters.

4 Experiments

Datasets. For our experiments, we use the Amazon 5-core[4] dataset on CDs, Toys, Music, Kindle, Electronics, Movies&TV and the Yelp 2018[5] dataset which are common benchmarks for recommendation systems. To preprocess the datasets, we perform tokenization, part-of-speech tagging and stemming with NLTK.[6] For the summary generation, we represent words using the Google News embeddings for English. Table 1 presents the statistics of the each dataset in terms of the number of reviews, users, items and vocabulary size.

[4] `http://jmcauley.ucsd.edu/data/amazon`
[5] `https://www.yelp.com/dataset/challenge`
[6] `https://www.nltk.org/`

Table 1. Basic statistics of evaluation dataset.

Dataset	Reviews	Users	Items	User Vocab	Item Vocab
CDs	1,097,592	75,258	64,443	363,883	418,414
Toys	167,597	19,412	11,924	56,456	59,414
Music	64,706	5,541	3,568	78,293	83,904
Kindle	982,619	68,223	61,934	184,885	205,915
Electronics	1,685,748	192,403	63,001	256,920	235,408
Movies	1,697,533	123,960	50,052	397,060	495,021
Yelp	3,072,057	199,445	115,798	335,831	340,526

Previous methods and baselines. We compare our rating prediction component to the following recommendation algorithms as baselines: Probabilistic Matrix Factorization (PMF) [21] is a Matrix Factorization method using Gaussian distribution to model the users and items latent factors. Non-negative matrix factorization (NMF) [10] factorizes the rating matrix into a user matrix and item matrix to have no negative elements. Singular Value Decomposition (SVD++) [9] is a collaborative filtering method which creates the latent factors considering implicit feedback information. Hidden Factors as Topics (HFT) [16] is a state-of-the-art method that combines latent rating dimensions with latent review topics using exponential transformation function to link the stochastic distributions. Deep Cooperative Neural Networks (DeepCoNN) [27] is a state-of-the-art method that jointly models users and items from textual reviews using two parallel neural networks coupled using a shared output layer. We also utilize the extended version DeepCoNN++, where the shared layer with the Factorization Machine estimator is replaced with a neural prediction layer. The Neural Attentional Regression with Reviews-level Explanation (NAARE) model [4] is a state-of-the-art method that uses a similar neural network architecture as DeepCoNN++, but additionally uses an attention-based review pooling mechanism to select the reviews for modeling.

Additionally, we compare our review summary generation component to multiple state-of-the-art unsupervised and supervised summarization methods: TF⋆IDF [15] selects sentences based on their term-frequency-inverse-document-frequency scores. LexRank [5] scores sentences based on PageRank. LSA [23] applies dimensionality reduction to the term-document matrix using singular value decomposition (SVD). KL-Greedy [8] minimizes the Kullback-Leibler (KL) divergence between the word distributions in the document and the summary. ICSI [6] is a global linear optimization method, which extracts a summary by solving a maximum coverage problem of the most frequent bigrams. Seq2Seq-gen [20] is a sequence-to-sequence encoder-decoder model, which encodes the input sequence using a bi-directional LSTM network and decodes using a conditional bi-directional LSTM network with attention. Finally, Pointer-gen denotes the sequence-to-sequence pointer-generator network by [22] using the pointer mechanism to determine a probability function to decide whether to generate the words from the vocabulary or copy from the source.

Experimental setup. We divide each of the datasets into training, development and testing consisting of 80%, 10%, and 10% of the data respectively, which is a typical split ratio in recommendation evaluation. For all baseline methods (PMF, NMF, SVD++, HFT[7]), we use the Librec toolkit[8] and select the number of latent factors for each domain after fine tuning on the development set. To calculate the topic vectors, we set the tf-idf vectors size to 1,000 and the number of topics k to 10. For our neural network based approach, after hyperparameter fine tuning using random search, we set the latent factors to 32 and the number of hidden layers to 2. For the gradient-based optimization, we use the Adam optimizer.

For review summary generation, we set the beam size to 10 and the maximum summary length to 10 as nearly 80% of the summaries have a maximum of 10 words. We randomly initialize the neural network parameters.

Evaluation Metrics. To evaluate the rating prediction component, we employ two widely used metrics for recommender systems: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE):

$$\text{MAE} = \sum_{u,v} \frac{|r_{u,v} - \hat{r}_{u,v}|}{n}, \qquad \text{RMSE} = \sqrt{\sum_{u,v} \frac{(r_{u,v} - \hat{r}_{u,v})^2}{n}}, \tag{15}$$

where $r_{u,v}$ is the ground-truth rating, $\hat{r}_{u,v}$ is the predicted rating for a given user u and item v pair, n is the total number of ratings between users and items. To evaluate the review summary generation component, we use ROUGE-1 (R1) and ROUGE-2 (R2) scores [13] between the generated summary and the gold standard summary using the common parameters `-c 95 -r 1000 -n 2 -a -m`.

5 Results and Analysis

In the following section we analyze the performance of our J3R system in terms of (a) the rating prediction component, (b) the review summary generation component, and (c) its capabilities to explain recommendations.

5.1 Rating Prediction Analysis

Table 2 shows the results of the rating prediction component in comparison to the baselines. It shows that our model J3R consistently outperforms all other methods in terms of MAE and RMSE scores on all datasets. We also observe that the collaborative filtering methods PMF and NMF have low performance scores compared to other baselines. In contrast, SVD++ shows that it is still a strong baseline for recommendation systems as shown in the Netflix Prize 2008.[9]

[7] `https://github.com/lipiji/HFT`
[8] `https://www.librec.net/dokuwiki/doku.php?id=Recommender`
[9] `https://www.netflixprize.com/community/topic_1537.html`

Table 2. MAE and RMSE scores (lower is better) for our models (lower group) in comparison to the state-of-the-art models (upper group). Best results are bold-faced. Italic and underlined results of J3R-Pointer are significantly better than NAARE, SVD++, and DeepCoNN++ with $p < 0.05$.

Models	CDs		Toys		Music		Kindle		Electronics		Movies		Yelp	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	.682	.972	.705	.979	.849	.922	.573	.835	.855	1.193	.765	1.083	.967	1.273
NMF	.749	1.082	.693	.999	.700	.997	.651	.956	.952	1.366	.830	1.176	1.024	1.381
SVD++	.667	.956	.636	.907	**.641**	.905	.540	.790	.848	1.163	.750	1.043	.953	1.236
HFT	.746	.979	.645	.892	.665	.911	.664	.869	.846	1.112	.838	1.076	1.028	1.252
DeepCoNN	.695	.944	.669	.912	.672	.901	.565	.791	.866	1.124	.750	1.016	.938	1.186
DeepCoNN++	.682	.933	.652	.900	.659	.894	.553	.783	.824	1.113	.742	1.002	.922	1.202
NARRE	.675	.930	.683	.906	.698	.925	.547	.785	.834	1.107	.736	1.001	.921	1.186
MLP	.751	.995	.695	.967	.710	.990	.627	.857	.875	1.167	.816	1.083	.997	1.324
MLPTopic	.706	.954	.674	.943	.685	.907	.602	.814	.839	1.113	.758	1.059	.967	1.258
J3R-Seq2Seq	.685	.937	.647	.899	.660	.892	.560	.794	.823	1.052	.746	1.008	.919	1.174
J3R-Pointer	***.661***	***.912***	***.634***	***.880***	.656	**.890**	***.538***	***.775***	***.805***	***.995***	***.714***	***.984***	***.881***	***1.009***

SVD++ performs on par or better in comparison to the state-of-the-art neural content-based systems like DeepCoNN, DeepCoNN++, and NARRE on small and medium-sized data. However, the neural approaches perform better on large datasets. Overall, the results show that our J3R-Pointer model performs better in terms of MAE and RMSE scores as compared to the best baseline methods NAARE and SVD++. This shows that review information helps in improving the representation of the user and item latent factors, which is further enhanced with the joint learning of rating prediction and review summary generation. The improvement is consistent and significant across the six datasets, whereas it is slightly lower on the Music dataset (−1.5%) compared to Electronics (+2.9%), Movies&TV (+2.2%), or Yelp (+4.0%). The lower scores for Music is due to fewer reviews available for content-based models, which explains that latent factors of SVD++ also capture better information when there is less training data.

Ablation analysis. To quantify the impact of each component on the rating prediction task, we do an ablation analysis. We try two different settings contrasting two single-task learning setups with our MTL setup: (a) *MLP*: the rating prediction component (section 3.2), where a multi-layer perceptron based rating prediction model is randomly initialized with user and item vectors, (b) *MLP-Topic*: the rating prediction component plus the topic vector component (section 3.1 and 3.2) and (c) two variants of our full setup including all three components and the multi-task learning framework to jointly predict ratings and generate review summaries using user and item topic vectors initialized by the LDA topic vectors: J3R-Pointer is our proposed method using the pointer-generator network. J3R-Seq2Seq is an alternative to [12], where the GRU layers are replaced with LSTM and the rating regression has three hidden layers instead of one.

Table 3. ROUGE-1 (R1) and ROUGE-2 (R2) precision scores of the generated summaries. Higher values are better. Best results per dataset are shown in bold.

Models	CDs		Toys		Music		Kindle		Electronics		Movies		Yelp	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
TF⋆IDF	.078	.017	.097	.027	.079	.019	.087	.024	.098	.029	.087	.023	.191	.126
LexRank	.087	.021	.107	.031	**.087**	**.024**	.097	.024	.109	.035	.096	.027	.204	.126
LSA	.068	.012	.077	.018	.068	.013	.070	.015	.081	.020	.074	.016	.122	.061
KL-Greedy	.070	.013	.080	.018	.073	.015	.074	.017	.086	.023	.078	.017	.141	.079
ICSI	.047	.010	.064	.017	.043	.008	.058	.017	.061	.018	.050	.012	.119	.064
Seq2Seq-gen	.108	.025	.114	.026	.053	.005	.139	.035	.177	.065	.134	.040	.219	.131
Pointer-gen	.135	.039	.122	.030	.059	.007	.152	.047	.179	.069	.141	.052	.250	.163
J3R-Seq2Seq	.119	.030	.120	.031	.060	.010	.150	.042	.185	.078	.145	.059	.235	.148
J3R-Pointer	**.156**	**.045**	**.137**	**.040**	.065	.012	**.185**	**.053**	**.190**	**.082**	**.159**	**.065**	**.274**	**.181**

Table 2 shows that *MLPTopic* performs better than the simple *MLP* model, which explains that the LDA topic vectors are useful for rating prediction as they capture user–item preferences. Our best performing model J3R-Pointer outperforms the individual components consistently across different domains. This elucidates that multi-task learning of rating prediction with review summary generation initialized with LDA based user and item models capture better user and item latent vectors. Furthermore, J3R-Pointer performs better than J3R-Seq2Seq and shows that the use of pointer network helps in better predictions.

5.2 Review Summary Generation Analysis

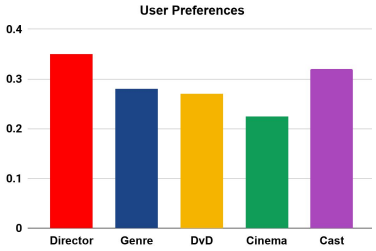
Although summarization is our auxiliary task to assist our main task of rating prediction, we separately evaluate the performance of our review summary generation component in this section. Table 3 shows the comparison of the review summary generation of J3R with baseline summarization models.

LexRank is the best-performing method among all the extractive baselines and performs the best on the Music dataset. However, the results show that the generative methods (i.e., Seq2Seq-gen and Pointer-gen) improve in ROUGE-1 and ROUGE-2 when compared to the baseline systems on the other six datasets, whereas for the Music dataset the results are only slightly lower than the best performing system LexRank. Our J3R-Pointer model performs best among all generative methods, exhibiting that the multi-task learning-based method captures user importance during summary generation. For the Music domain, we observe that the generative methods perform worse than the extractive methods due to the small data size available for training. Another reason is that J3R-Pointer's pointer-generator network tends to produce short abstractive summaries, while the extractive baselines produce longer summaries increasing the chances of overlaps with the gold summary. Furthermore, from the data analysis across datasets we observe that about 30% of the dataset have zero ROUGE-1

Table 4. Top five words for each of the top five topics of Movie&TV (left) and Yelp restaurant domain (right) explained with the most representative words.

Director	Genre	DVD	Cast	Cinema
seasons	story	video	actor	scene
episodes	style	collection	role	family
part	horror	quality	performance	love
point	comedy	television	voice	relationship
release	drama	series	dialogue	experience

Food	Service	Cuisine	Breakfast	Price
restaurant	server	greek	egg	check
main course	menu	chinese	eat	pay
taste	time	pizza	pancake	money
experience	owner	rice	sandwich	stay
soup	stay	ramen	fresh	cost



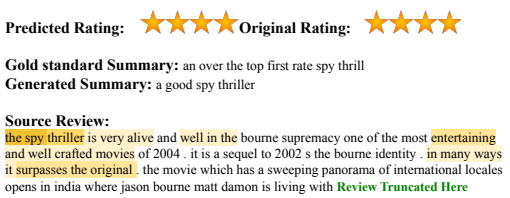


Fig. 3. (left) Interpretation of the user preferences using an histogram over top five topics from the topic model. (right) Word importance on the source review shows the evidence for the predicted rating.

and ROUGE-2 scores, which explains the overall low ROUGE-1 and ROUGE-2 across various methods.

5.3 Explainability Analysis

Besides performance improvements, an important advantage of our J3R system is the interpretability of the generated recommendations. In this section, we analyze two ways of explanations: (a) illustrating the importance of different topics with respect to a user based on topic vectors and (b) illustrating the word importance in the reviews while summarizing the content for the user.

First, our user model described in Section 3.1 illustrates the user's preferences on the important aspects of a domain. Table 4 shows the top five topics with their most representative word and the top five words describing each topic in the Movies&TV and the Yelp restaurant domain. To gain a better interpretation of the topic words, we remove words belonging to multiple topics. Thus, based on the topic distribution θ_d of important words in a domain and the distribution

of the words $\phi_{z_{dn}}$ across a topic, a user's preferences are computed from the user vector u created from the reviews written by the user. An example explanation of the preferences of a user who has written 490 reviews in Movies&TV is shown in the histogram on the left-hand side of Figure 3.

Second, we use the representative words in a review as evidence to explain the rating. We investigate word importance in the review using the attention weights. Figure 3 illustrates an example from the Movie&TV domain on Jason Bourne (2004). In the figure, we describe a scenario where the user decides to buy the DVD of Jason Bourne (2004). The user is overwhelmed by hundreds of reviews before making up the mind about the movie. Our J3R model summarizes each review and illustrates the most representative words of the review using the attention mechanism as described in Section 3.3. On the right-hand side of figure 3, we highlight the word importance in the source review based on attention weights while generating a review summary. The example shows that phrases like "the spy thriller", "entertaining", "surpasses the original" are highlighted by our model for the generated summary "a good spy thriller". Furthermore, the generated summary and the gold standard summary illustrate the same aspects of a movie (e.g., "genre", "director style").

6 Conclusion and Future Work

We propose a novel explainable recommendation system J3R using an MTL approach to jointly model user rating prediction and review summary generation. Our review summary generation model uses a pointer-generator network with an attention-based framework on user preferences and product attributes extracted from review content and user ratings as context vectors to generate a summary text, which in turn is used as evidence for explaining the predicted rating. Empirical results on benchmark datasets show that J3R achieves better performance than the state-of-the-art models on both rating prediction as well as review summary generation.

In future work, we plan to investigate the cold-start problem, since our model performs well when there is enough information about the users from the review content. However, when there is a new user or a user with less reviews, it is difficult to estimate the user preferences. Thus, a neighborhood model to calculate the similarity of the user to existing users and preference forms can estimate the user preferences with respect to the items. Similarly, for a new product the product attributes similarity can be used to initialize the latent factors for the rating prediction component and the review summary generation component. Furthermore, it would be interesting to explore sentiment analysis as a multi-task approach which is similar to rating prediction.

Acknowledgments

This work has been supported by the German Research Foundation as part of the Research Training Group "Adaptive Preparation of Information from

Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1. Zhifeng Bao and Mark Sanderson are supported by ARC DP170102726.

References

1. Aciar, S., Zhang, D., Simoff, S.J., Debenham, J.K.: Informed recommender: Basing recommendations on consumer product reviews. IEEE Intelligent Systems **22**(3), 39–47 (2007)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. In: Proceedings of NIPS. pp. 601–608. Vancouver, Canada (2001)
3. Catherine, R., Mazaitis, K., Eskenazi, M., Cohen, W.W.: Explainable entity-based recommendations with knowledge graphs. In: Proceedings of RecSys. Como, Italy (2017)
4. Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: Proceedings of WWW. pp. 1583–1592. Lyon, France (2018)
5. Erkan, G., Radev, D.R.: LexRank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research **22**, 457–479 (2004)
6. Gillick, D., Favre, B.: A scalable global model for summarization. In: Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing. pp. 10–18. Boulder, CO, USA (2009)
7. Goodman, B., Flaxman, S.R.: European union regulations on algorithmic decision-making and a "right to explanation". AI Magazine **38**(3), 50–57 (2017)
8. Haghighi, A., Vanderwende, L.: Exploring Content Models for Multi-document Summarization. In: Proceedings of NAACL. pp. 362–370. Boulder, CO, USA (2009)
9. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of SIGKDD. pp. 426–434. Las Vegas, NV, USA (2008)
10. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Proceedings of NIPS. pp. 556–562. Denver, CO, USA (2000)
11. Li, P., Wang, Z., Bing, L., Lam, W.: Persona-aware tips generation. In: Proceedings of WWW. pp. 1006–1016. San Francisco, CA, USA (2019)
12. Li, P., Wang, Z., Ren, Z., Bing, L., Lam, W.: Neural rating regression with abstractive tips generation for recommendation. In: Proceedings of SIGIR. pp. 345–354. Shinjuku, Tokyo, Japan (2017)
13. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of the ACL-04 Workshop on Text Summarization Branches Out. pp. 74–81. Barcelona, Spain (2004)
14. Liu, B.: Sentiment analysis and subjectivity. In: Handbook of Natural Language Processing, Second Edition., pp. 627–666. Boca Raton: CRC Press (2010)
15. Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of Research Development **2**, 159–165 (1958)
16. McAuley, J.J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of RecSys. pp. 165–172. Hong Kong, China (2013)
17. Mukherjee, S., Popat, K., Weikum, G.: Exploring latent semantic factors to find useful product reviews. In: Proceedings of the 2017 SIAM International Conference on Data Mining. pp. 480–488. Houston, TX, USA (2017)

18. Musat, C.C., Liang, Y., Faltings, B.: Recommendation using textual opinions. In: Proceedings of IJCAI. pp. 2684–2690. Beijing, China (2013)
19. Rei, M.: Semi-supervised multitask learning for sequence labeling. In: Proceedings of ACL. pp. 2121–2130. Vancouver, Canada (2017)
20. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In: Proceedings of EMNLP. pp. 379–389. Lisbon, Portugal (2015)
21. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Proceedings of NIPS. pp. 1257–1264. Vancouver, Canada (2007)
22. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: Proceedings of ACL. pp. 1073–1083. Vancouver, Canada (2017)
23. Steinberger, J., Jezek, K.: Using latent semantic analysis in text summarization and summary evaluation. In: Proceedings of the 7th ISIM Conference. pp. 93–100. Rožnov pod Radhoštěm, Czech Republic (2004)
24. Tintarev, N., Masthoff, J.: Designing and evaluating explanations for recommender systems. In: Recommender Systems Handbook, pp. 479–510. Springer (2011)
25. Yu, N., Huang, M., Shi, Y., Zhu, X.: Product review summarization by exploiting phrase properties. In: Proceedings of Coling. pp. 1113–1124. Osaka, Japan (2016)
26. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: Proceedings of SIGIR. pp. 83–92. Gold Coast, Australia (2014)
27. Zheng, L., Noroozi, V., Yu, P.S.: Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of WSDM. pp. 425–434. Cambridge, UK (2017)
28. Zhou, M., Lapata, M., Wei, F., Dong, L., Huang, S., Xu, K.: Learning to generate product reviews from attributes. In: Proceedings of EACL. pp. 623–632. Valencia, Spain (2017)