Accurate and Explainable Recommendation via Review Rationalization

Sicheng Pan*
School of Computer Science
Fudan University
Shanghai, China
scpan15@fudan.edu.cn

Dongsheng Li
Microsoft Research Asia
Shanghai, China
dongsli@microsoft.com

Hansu Gu†
Seattle, United States
hansug@acm.org

Tun Lu*†
School of Computer Science
Fudan University
Shanghai, China
lutun@fudan.edu.cn

Xufang Luo
Microsoft Research Asia
Shanghai, China
xufang.luo@microsoft.com

Ning Gu*
School of Computer Science
Fudan University
Shanghai, China
ninggu@fudan.edu.cn

ABSTRACT

Auxiliary information, such as reviews, have been widely adopted to improve collaborative filtering (CF) algorithms, e.g., to boost the accuracy and provide explanations. However, most of the existing methods cannot distinguish between co-appearance and causality when learning from the reviews, so that they may rely on spurious correlations rather than causal relations in the recommendation — leading to poor generalization performance and unconvincing explanations. In this paper, we propose a Recommendation via Review Rationalization (R3) method including 1) a rationale generator to extract rationales from reviews to alleviate the effects of spurious correlations; 2) a rationale predictor to predict user ratings on items only from generated rationales; and 3) a correlation predictor upon both rationales and correlational features to ensure conditional independence between spurious correlations and rating predictions given causal rationales. Extensive experiments on real-world datasets show that the proposed method can achieve better generalization performance than state-of-the-art CF methods and provide causal-aware explanations even when the test data distribution changes.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

recommendation; rationalization; explainability

*Also with Shanghai Key Laboratory of Data Science, Fudan University, Shanghai, China.
†Corresponding author.

ACM Reference Format:
Sicheng Pan, Dongsheng Li, Hansu Gu, Tun Lu, Xufang Luo, and Ning Gu. 2022. Accurate and Explainable Recommendation via Review Rationalization. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512029

1 INTRODUCTION

In recent years, researchers have been applying state-of-the-art deep learning techniques to improve performance in recommender systems, for both rating prediction [5, 8, 21, 38, 48] and top-n recommendation [11, 15, 40, 41]. A large body of recent works exploits rich auxiliary information as complements to improve the performance of recommender systems. One of the promising and popular directions is to leverage the rich textual information in user reviews. Firstly, many CF methods [5, 8, 11, 38, 48] were proposed to extract features from user reviews and item reviews through deep learning architectures, and then use these features to perform more accurate rating prediction [5, 8, 38, 48] or top-n recommendation [11]. Besides accuracy, the review information can also help enhance explainability in recommendation [46]. Several review-based recommender systems utilize variants of attention mechanism [39], which can assign different weights to words in reviews and then provide explanations about informative levels of words for recommendation. However, neural models may "cheat" by exploiting spurious correlations in the training data, which results in high reported accuracy or good explanation during training but can lead to low generalization performance when data distributions change during test time [19].

Firstly, spurious correlations may hurt the accuracy of recommender systems. Selection bias [23] commonly exists when collecting data for training recommendation algorithms. Thus, CF algorithms may rely on spurious correlations between users/items with few observed ratings [22, 23] and even amplify these biases [3, 42] in model learning, which may lead to unsatisfactory performance when data distribution changes. Secondly, spurious correlations may lead to unconvincing explanations in machine learning applications including recommender systems. For instance, a recent work [31] set up an experiment using neural networks for feature engineering to distinguish pictures between Wolves and Eskimo

Dogs (huskies). They hand-selected the training set to ensure that all pictures of wolves had snow in the background, while pictures of huskies did not. Interestingly, they found that the classifier will predict "Wolf" if there is snow (or light background), and "Husky" otherwise, regardless of animal color, position and pose. In this example, explaining "wolf" by only snow is obviously wrong. Similarly, spurious correlations raise challenges for explanation in recommender systems. Xu et al. [43] mentioned that many existing recommender systems use global association rule mining to discover relationships among items and rely on the item co-occurrence for explanation, which may lead to non-personalized explanations because different users would receive the same explanation as long as they are recommended with the same item and have overlapped histories. Therefore, it is necessary to eliminate these spurious correlations when training CF models and explaining recommendations.

Inspired by Structural Causal Model (SCM) [29], we design a framework based on d-separation to extract rationale from textual reviews to achieve accurate and explainable recommendation to tackle the above challenges. We refer rationales to the features R who can make all the other features C being independent of the predicting results Y, when given R. We propose the Recommendation via Review Rationalization (R3) method consisting of: 1) a rationale generator, which first generates potential rationales from reviews in word level and then refines the rationales by matching them with user preferences and item properties; 2) a rationale predictor, which can predict user ratings on items only from generated rationales via a neural form of latent factor model; and 3) a correlation predictor, which is built upon both rationales and correlational features to ensure conditional independence between spurious correlations and rating predictions given rationales.

The main contributions of this work are summarized as follows:

- We show that accurate and explainable recommendation can be approached from a causal perspective and propose a novel review rationalization method to extract rationales based on d-separation from reviews.
- We propose the Recommendation via Review Rationalization (R3) method which can fully utilize the rationales from reviews to boost recommendation accuracy and provide causal-aware explanations.
- We evaluate the proposed method on real-world datasets with and without data distribution shifts and the empirical results demonstrate that R3 can extract rationales which are generic with varying data distribution or even across datasets and achieve higher accuracy.

2 PRELIMINARIES

2.1 Motivating Example

Let us consider a scenario that users rely on online applications to find restaurants, e.g., Yelp. Recommender systems can be built to provide recommendations to users. However, these recommendations may not contain obvious indications of why such recommendations are provided. To answer "why" recommendations are made, we can find a small subset of words from the reviews, namely rationales, that suffices on its own to make the recommender system yield the same outcome. The following review is from Yelp and the boldface words are extracted using OpinionDigest [36] to summarize abstractive opinion:

> **Noodles** are truly **great**! The **meat sauce** and **lamb** are my **favorites**! This is a perfect to-go place near the freeway for a work day including weekends! :-)

Aspects (e.g., noodles) and opinions (e.g., great) from the above review may be strong enough for recommender system to bring up a positive recommendation. This case is not isolated and hints at an intuitive approach: we can design a selector to find aspects and opinions from the reviews and build a predictor for rating prediction based on the aspects and opinions. Such approach may exhibit competitive accuracy and help to explain the recommendation. However, it may also be trapped by spurious correlations. For instance, the recommender system may select "weekday fast food" as a strong feature for recommendations, which could be a spurious correlation for a user who values taste more than convenience.

We expect our model to extract the most informative features for prediction, which means that other features cannot provide more information when giving them. We refer such features as rationales R, who can make all the other features C being independent of the predicting results Y, when given R. This leads us to Structural Causal Model (SCM) [29], where we can apply d-separation to find such features.

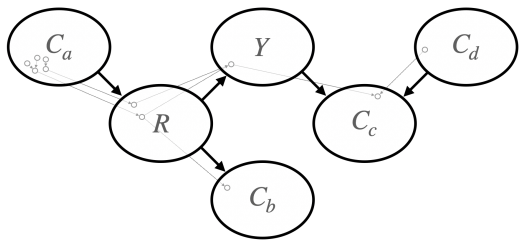

Figure 1: An abstracted causal graph for rating prediction. Y represents the ratings, R represents the rationales and C_* represents all the features correlated to Y but not rationales.

2.2 Causal Graph

We propose to extract rationales R, as shown in Figure 1, from all input features R and C_* (the set of all correlations) to predict the rating Y. Since the directed edge $R \longrightarrow Y$ denotes that R is the direct cause of Y, rationale features should be more robust for prediction among different scenarios, e.g., varying data distributions.

Figure 1 shows an abstracted causal graph for the rating prediction task, where Y denotes the ratings, R denotes the rationales and C_* denotes all the correlations except rationales. For instance, R could be the descriptions of food quality of a restaurant, and C_* could be the correlation features like the location of a restaurant or the gender of the user who posted the review. Figure 1 contains three basic graphical building blocks, i.e., chain $C_a \longrightarrow R \longrightarrow Y$, fork $C_b \longleftarrow R \longrightarrow Y$, and immorality $Y \longrightarrow C_c \longleftarrow C_d$. According to d-separation, we know that

$$Y \perp C_a, C_b, C_d | R, \tag{1}$$

which means that C_* cannot provide extra information beyond R to predict Y.

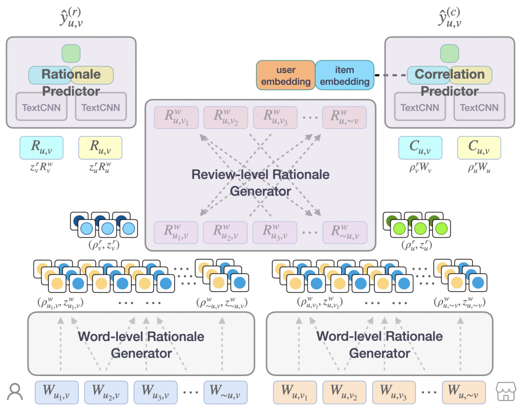
Rationale Predictor
TextCNN
TextCNN
user embedding
item embedding
Correlation Predictor
Review-level Rationale Generator
Word-level Rationale Generator
Word-level Rationale Generator

Figure 2: The architecture of R3. R3 uses review content ($W_{u,*|\sim t_{u,v}}$, $W_{*,v|\sim t_{u,v}}$) to automatically extracts rationales R_* via the proposed rationale generator and calculates correlations C_* of reviews. Then, R3 provides rating predictions based on these two types of features separately.

Rationales can be extracted via constructing a minimal feature set with maximum predictivity from the restricted reviews. Here, the maximum predictivity means that all other variables are independent of the target variable when conditioned on this feature set. Generally, under certain assumptions [28], such minimal feature set is the Markov boundary of the target variable Y [28]. Since the directs effects (C_c) are not helpful in the recommendation tasks, we can simply add a restriction on our model that we only use reviews generated before to predict the current rating. Since an effect cannot occur before its cause, effects related features cannot be extracted from the previous reviews as rationales, and thus we cannot observe and condition on C_c by following the restriction. Then, C_d needs not to be conditioned on because of the immorality structure, i.e., C_d and Y are independent when C_c is not observed.

In this work, the rationales are defined as the minimal feature set with maximum predictive ability extracted from previous reviews to predict the current rating.

2.3 Problem Formulation

We focus on the rating prediction task and formulate the targeted problem as follows:

Input: The input of our approach includes a user set U, an item set V, and a corpus of user-item interactions D including ratings and reviews.

- Each user is represented by its user ID $u \in U$ and each item is represented by the item ID $v \in V$.
- Each user-item interaction in D is denoted as a 5-tuple $(u, v, W_{u,v}, y_{u,v}, t_{u,v})$, where $W_{u,v}$ is the textual content of user u's review on item v, $y_{u,v}$ is the associated rating, and $t_{u,v}$ is the timestamp of the review. The review corpus of u is denoted by $\{W_{u,v_1}, W_{u,v_2}, \ldots, W_{u,v_{N_u}}\}$, where all N_u reviews of u are sorted by timestamp in ascending order. The review corpus of v is constructed similarly.

Output: Given a user u, an item v, and the corresponding review corpus $W_{u,*|\sim t_{u,v}} = \{W_{u,v_1}, \ldots, W_{u,\sim v}\}$ and $W_{*,v|\sim t_{u,v}} = \{W_{u_1,v}, \ldots, W_{\sim u,v}\}$, where $W_{u,\sim v}$ means the latest item review of user u before user u commented on item v, and $W_{\sim u,v}$ means the latest user review of item v before user u commented on item v. Our goal is to:

- Extract the rationales R from reviews that is the direct cause of the rating Y.
- Predict the rating $\hat{y}_{u,v}$ that reflects how much u likes v.

3 R3: RECOMMENDATION VIA REVIEW RATIONALIZATION

As shown in Figure 2, the proposed recommendation via review rationalization (R3) consists of three key components:

- **Rationale generator**, which consists of a word-level rationale generator and a review-level rationale generator. The word-level rationale generator predicts the probability of each word being rationale (by $\rho^w_{u,v}$) and binarizes the probabilities to select the top words from each review (by $z^w_{u,v}$). After that, the review-level rationale generator selects rationale reviews based on the word-level rationales.
- **Rationale Predictor**, which only takes rationales as input to predict user ratings on items. The rationale in our definition corresponds to the selected words in each review, i.e., $\{w_k|z_k = 1\}$. The rationale predictor first embeds input rationales into vectors and then uses a neural form of latent factor model to predict the ratings.
- **Correlation Predictor**, which takes all features as input including both rationales and correlated features. Since the rationale generator provides the probability of each word w_t to be selected as rationales, i.e., $\{\rho_1, \ldots, \rho_l\}$, we can use the probabilities to weigh the importance of words in the predictor. The rest of it is similar to the rationale predictor.

The above two predictors are designed to ensure that the extracted rationales are indeed the set of features with maximum predictive ability by minimize the gap between their losses.

3.1 Rationale Generator

We design the rationale generator by a two-level approach to extract both word-level and review-level rationales. We process text using the same approach as the text processor in existing state-of-the-art review-based CF methods, e.g., DeepCoNN [48] and NARRE [8].

3.1.1 Word-level rationale generator. In the word-level rationale generator, we first pass the historical reviews of user u and item v into the text processor. Then, the features obtained by the text processor can be used to calculate the probability of each word being rationale. The output is concatenated over the neurons and then passed to a fully connected layer consisting of a weight matrix $W \in \mathbb{R}^{md\times 1}$ and a bias b as follows:

$$\rho = \sigma(cW + b), \tag{2}$$

where c is the concatenated feature from the output of the text processor for each word, σ is the sigmoid activation function, and $\rho_{u,v}$ represents the probabilities of the words being selected as rationales from the review that user u gave to item v.

To specifically choose rationales, we transform probabilities ρ to binary signals denoted by z as follows:

$$z = \rho + f_d(\lfloor \rho \rceil - \rho). \tag{3}$$

$\lfloor \cdot \rceil$ is the rounding function and f_d is the detach function indicating that no gradients are needed. Intuitively, z is a binary mask to select rationale words from each review based on the probabilities ρ.

3.1.2 *Review-level rationale generator.* After obtaining word-level rationales, we further identify rationales that fully reflect user preferences and item properties for each targeted user-item pair. We model this as interactions among four information: user preference $\mathcal{P}_u$, word-level rationales from user historical reviews $R^w_{u,*}$, item property $\mathcal{P}_v$, and word-level rationales from item historical reviews $R^w_{*,v}$.

Intuitively, user u reveals its preference on item v by posting a review containing a set of word-level rationales denoted as $R^w_{u,v}$. We can first model user preference by its review ($P(p_{u,v}|R^w_{u,v})$) using the text processor. After obtaining features generated by the text processor, we apply aggregation over the dimension of reviews as follows:

$$o_j = \max(c^1_j, \dots, c^l_j),\ O = [o_1, \dots, o_m], \tag{4}$$

where c_j is the output of the convolutional layer in the text processor, and c^t_j is the feature produced by the j-th neuron from the sliding windows t over the embedding matrix. Similarly, we obtain features from item rationales $R^w_{u,v}$ in the same way. Then, we pass the outputs from user reviews and item reviews to a fully connected layer to model user preference on items $p_{u,v}$.

After modelling the user preferences, we obtain the user preference vector $\mathcal{P}_u = \{p_{u,1}, \dots, p_{u,|W_{u,*|\sim t_{u,v}}|}\}$ reflecting user interests and the item preference vector $\mathcal{P}_v = \{p_{1,v}, \dots, p_{|W_{*,v|\sim t_{u,v}}|,v}\}$ reflecting item properties. Then, we calculate an affinity matrix between $\mathcal{P}_u$ and $\mathcal{P}_v$, to select rationales that match both user interests and item properties:

$$s_{u,v} = \mathcal{P}_u^\top \mathcal{P}_v. \tag{5}$$

By taking the row and column wise sum of the affinity matrix s and using them to weigh the original list of reviews $W_{u,*|\sim t_{u,v}}$ and $W_{*,v|\sim t_{u,v}}$, we are able to obtain the probabilities of potential rationales to become true rationales. Based on the affinity matrix s, we compute the probabilities of user/item reviews being rationales as follows:

$$\rho^r_u = \sigma(\sum_{row} s),\ \rho^r_v = \sigma(\sum_{col} s), \tag{6}$$

where ρ^r_u or ρ^r_v is the probabilities of user or item reviews to be rationale reviews, respectively. Note that the review-level rationales only contain the rationale words from the word-level rationale generator.

3.2 Rationale Predictor

The rationale generator provides us with two levels of masks for rationales, with which we can select rationale reviews containing only rational words. Based on the final rationales, we can build the rationale predictor to predict user ratings on items only by rationale features.

We first use the text processor to learn user preference and item properties from the rationales. We pass the rationales of user u to the text processor to get the user preferences matrix $\mathcal{P}_u \in \mathbb{R}^{d \times l}$, where d is the output size of the final fully connected layer and l is the number of historical reviews from u. We then binarize the probabilities from the rationale generator (by Equation 3) to obtain the masks for rationales (denoted by z^r_u) and apply a binary weighted sum to represent the final user rationale features γ^r_u. Similarly, the final item rationale features γ^r_v can be obtained as follows:

$$\gamma^{(r)}_u = {z^r_u}^\top \mathcal{P}_u,\ \gamma^{(r)}_v = {z^r_v}^\top \mathcal{P}_v. \tag{7}$$

Finally, we use a neural form of latent factor model to predict the user ratings on items as follows:

$$\hat{y}^{(r)}_{u,v} = h_r([\gamma^{(r)}_u, \gamma^{(r)}_v]) + b_u + b_v + \mu, \tag{8}$$

where μ is the global rating bias, b_u is the user rating bias, b_v is the item rating bias, and h_r is a fully connected layer.

3.3 Correlation Predictor

Despite the rationale indicator mask, the rationale generator also provides the probability of each word being selected as rationale. By applying such probabilities to the input reviews, we can build the correlation predictor to predict user ratings on items by utilizing both rationale features and non-rationale features.

Similar to rationale predictor, we first use the text processor to embed all the reviews and then apply a weighted sum to obtain user/item correlation features. Different from the rationale predictor, we apply rationale probabilities as weights here instead of binary weights as follows:

$$\gamma^{(c)}_u = {\rho^r_u}^\top \mathcal{P}_u, \quad \gamma^{(c)}_v = {\rho^r_v}^\top \mathcal{P}_v. \tag{9}$$

Finally, we add the interaction correlation model via matrix factorization [18] into the neural form of latent factor model to obtain the final correlation predictor as follows:

$$\hat{y}^{(c)}_{u,v} = h_c([\gamma^{(c)}_u + \gamma^{(e)}_u, \gamma^{(c)}_v + \gamma^{(e)}_v]) + b_u + b_v + \mu, \tag{10}$$

where $\gamma^{(e)}_u$ and $\gamma^{(e)}_v$ are the user embedding and the item embedding trained by matrix factorization.

3.4 Model Learning

We apply mean square loss for both rationale predictor and correlation predictor as follows:

$$\mathcal{L}_R = \sum_{u,v \in \mathcal{D}} (\hat{y}^{(r)}_{u,v} - y_{u,v})^2,\ \mathcal{L}_C = \sum_{u,v \in \mathcal{D}} (\hat{y}^{(c)}_{u,v} - y_{u,v})^2, \tag{11}$$

where $\mathcal{D}$ denotes the set of instances for training, and $y_{u,v}$ is the true rating assigned by the user u to the item v.

Since the rationale generator needs to construct a minimal feature set with maximum predictive ability, to satisfy the constraints introduced by d-separation in Figure 1, we add two terms to the optimization objective. The first term is to minimize the gap between $\mathcal{L}_R$ and $\mathcal{L}_C$ as follows:

$$\mathcal{L}_{R,C} = \mathbf{ReLU}(\mathcal{L}_R - \mathcal{L}_C), \tag{12}$$

which can help to ensure that features extracted by the rationale generator is as predictive as the case when all correlations are considered, i.e., ensure that the rationales R have the maximum predictive ability. The second term ensures that the size of the selected rationales is small via a sparsity constraint as follows:

$$\mathcal{L}_{Reg} = \mathbb{E}[||R||_1] - \gamma, \tag{13}$$

where γ is a predefined gap level and $\mathbb{E}[||R||_1]$ denotes the proportion of rationales in all input features. Finally, we design the optimization objective of the rationale generator as follows:

$$\mathcal{L}_G = \mathcal{L}_R + \lambda \mathbf{ReLU}(\mathcal{L}_R - \mathcal{L}_C) + \alpha(\mathbb{E}[||R||_1] - \gamma). \tag{14}$$

We apply a loop training strategy for this adversarial learning to optimize $\mathcal{L}_R$, $\mathcal{L}_C$ and $\mathcal{L}_G$.

4 EXPERIMENT

In this section, we conduct comprehensive experiments to answer the following research questions (RQs):

- **RQ1** Does R3 outperform state-of-the-art methods?
- **RQ2** Does R3 get rationales instead of correlations?
- **RQ3** What are the impacts of the design choices of R3?
- **RQ4** Does the rationales help improve explainability?

4.1 Experimental Setup

4.1.1 Dataset. We use five publicly available datasets from different domains to evaluate the proposed method. The first four datasets are from Amazon[1]: Home and Kitchen (35,515 users, 11,843 items, 341,138 reviews), Toys and Games (19,385 users, 11,912 items, 167,328 reviews), Health and Personal Care (38,577 users, 18,520 items, 346,089 reviews), and Beauty (22,348 users, 12,095 items, 198,378 reviews). Note that Beauty is only used to evaluate the performance of knowledge transfer across datasets because Beauty contains items (e.g., facial cream) that are similar to those in Health and Personal Care (e.g., after sun moisture), i.e., we train R3 on the Health-and-Personal-Care dataset and evaluate its performance on the Beauty dataset to evaluate the transferability of R3.

The Yelp[2] dataset is provided by OpinionDigest [36], in which aspects and opinions from reviews are extracted by Snippext [25]. Since the raw data is very large and sparse, we only kept the users and items with at least five ratings. After pre-processing, it contains 341,138 reviews from 35,515 users on 11,843 items.

We use randomized 80:10:10 train/validation/test splits. For each example, we ensure that user/item historical reviews are posted before the targeted timestamp. To evaluate the performance of R3 with data distribution shifts, we sample 2% of users whose average rating are around 3, which are much less biased than the original datasets, and we call them "debiased datasets". The reason why we choose 2% of users is that the number of their reviews is approximately equal to the size of the test sets. These debiased test sets are referred to as "test-u" in the experiments.

4.1.2 Evaluation Metrics. To evaluate the performance of all algorithms, we calculate Mean Square Error (MSE), which is widely used for rating prediction in recommender systems. A lower MSE score indicates a better performance. Given a predicted rating $\hat{y}_{u,v}$ and a ground-truth rating $y_{u,v}$ from the user u for the item v, the MSE is calculated as follows:

$$\mathbf{MSE} = \frac{1}{N}\sum_{u,v}(\hat{y}_{u,v} - y_{u,v})^2. \tag{15}$$

To evaluate the influence of biases in rating prediction, we apply Pearson Correlation Coefficient (PCC) between predicted ratings and the true ratings as follows:

$$\mathbf{PCC} = \frac{\sum_{u,v}(\hat{y}_{u,v} - \bar{\hat{y}}_{u,v})(y_{u,v} - \bar{y}_{u,v})}{\sqrt{\sum_{u,v}(\hat{y}_{u,v} - \bar{\hat{y}}_{u,v})^2}\sqrt{\sum_{u,v}(y_{u,v} - \bar{y}_{u,v})^2}}, \tag{16}$$

The intuition behind this is that predicted ratings with a constant bias may increase/decrease the MSE but will not affect the PCC. Hence we can evaluate the performance from a new perspective, i.e., if the performance improvements of R3 are from data biases or not.

To evaluate the performance of the proposed rationale generator, we compare the extracted rationales with the aspects and opinions extracted by Snippext [25] by the Explainability Recall measure [1]. Intuitively, aspects and corresponding opinions provided by users are the causes of the ratings. A higher recall indicates that our rationale generator extracts more of the aspects and opinions from the reviews. Since multiple words may have the same semantic meaning, we apply word vector cosine similarity to define synonyms. Extracting a synonym word can be treated as True Positive. We will evaluate the recall by applying different thresholds of synonym.

4.1.3 Baseline Methods. We consider a variety of methods following [32], which cover multiple categories of CF algorithms from traditional MF to state-of-the-art review-based deep-learning methods as follows. **1) MF** [18], in which ratings are modeled as: $\hat{y}_{u,v} = \gamma_u\gamma_v + b_u + b_v + \mu$. **2) D-CoNN** [48], which learns item properties and user behaviors jointly from reviews and models the rating by a neural network conditioned on the extracted features. An improved version named **D-CoNN++** (not in the original paper) is considered where the global, user, and item biases (μ, b_u, b_v) are learnt and added to the output. **3) T-Nets** [5], which uses the current review for regularization by minimizing the distance between the latent spaces. Similarly, an improved version named **T-Nets++** is consider in this paper where latent factors from MF are concatenated to the textual features. **4) MPCN** [38], which proposes a review-by-review pointer-based learning scheme to infer the importance of each review. **5) NARRE** [8], which follows the intuition of MPCN but uses an attention mechanism to learn the weights over individual reviews. **6) AHN** [13], which accounts for the difference of users' heterogeneous reviews and item's homogeneous reviews by means of asymmetric attentive modules. **7) ESCOFILT** [30], which integrates BERT, K-Means embedding clustering, and multilayer perceptron to learn sentence embeddings, representation-explanations, and user-item interactions, respectively.

We also compare our method with state-of-the-art explainable method **AR** [27], which extracting explanations from latent factor based recommendation algorithms by training association rules on the output of a matrix factorization-based black-box model, on the evaluation of explainability.

4.1.4 Implementation Details. For the compared methods, we reuse the implementations from the original papers. The hyper-parameters for the compared methods are initialized as those in the corresponding papers, and then carefully fine-tuned via grid search (with the same search space as our method) to achieve optimal performance. More specifically, we search the learning rate in [0.001, 0.005, 0.01, 0.05], L2 regularization coefficient in [10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}],

[1]https://cseweb.ucsd.edu/ jmcauley/datasets.html
[2]https://github.com/megagonlabs/opiniondigest

Table 1: Performance comparison (mean square error) on benchmark datasets. R3-R means the results of the rationale predictor and R3-C means the results of the correlation predictor. The best performance is in boldface. * and ** denote the statistical significance for $p < 0.05$ and $p < 0.01$, respectively, of R3-R compared to the best baseline.

(a) Raw Dataset	MF	D-CoNN	D-CoNN++	T-Nets	T-Nets++	MPCN	NARRE	AHN	ESCOFILT	R3-R	R3-C
Home and Kitchen	0.8882	0.8770	0.9895	0.9318	0.8952	0.8881	0.8815	0.8768	0.8778	**0.8421****	0.8432
Toys and Game	0.6916	0.6799	0.6770	0.682	0.6859	0.6925	0.6862	0.6806	0.6899	0.6760*	**0.6537**
Health and Personal Care	0.9239	0.9188	1.0332	0.9712	0.9404	0.9531	0.9626	0.8949	0.8798	0.8510**	**0.8470**
Yelp	1.0811	1.0796	1.0761	1.0715	1.0903	1.1097	1.0672	1.0774	1.0691	**1.0603****	1.0638

(b) Debiased Dataset	MF	D-CoNN	D-CoNN++	T-Nets	T-Nets++	MPCN	NARRE	AHN	ESCOFILT	R3-R	R3-C
Home and Kitchen	3.9481	3.8327	4.4353	3.8859	3.7996	4.3538	3.8801	3.6787	3.7414	**3.0957****	3.1048
Toys and Game	2.8575	2.8244	3.0766	3.0968	2.4824	2.7924	2.9847	2.3808	2.3613	**2.0158****	2.1809
Health and Personal Care	3.7198	3.8870	4.5853	3.5987	3.4528	4.1551	4.1778	3.3892	3.6773	**2.8640****	2.9443
Yelp	2.6131	2.8038	2.8559	3.014	2.6519	3.1663	2.8603	2.5824	2.6333	2.3466**	**2.3432**

and dropout rate in [0, 0.2, 0.4, 0.6, 0.8]. We use Adam [17] to adaptively change the learning rates for all the training. We train all models with early stopping (i.e., we stop model training if model performance does not improve for 10 epochs), and report the test results from the best performing models on the validation sets. In text embedding layer, we use GloVe[3] with the dimension of 50.

4.2 Experimental Results

4.2.1 Overall Performance (RQ1). Table 1 (a) compares the accuracy of R3 with all baseline methods in the standard rating prediction setting. We can observe that R3 outperforms all the compared methods on all four datasets, which confirms the superiority of our method in recommendation accuracy and answers RQ1.

Besides, we have an additional observation from Table 1 (a). Most of the methods considering reviews perform better than MF which only considers the rating matrix as the input. This is not surprising because review information is complementary to ratings and can be used to improve the quality of representation learning in CF. However, we also observe that, in a few cases, MF outperforms some of the review-based methods. We conjecture that this behaviour is mainly due to the spurious correlations applied by review-based methods. Since the parameter size of each review-based model is much larger than the MF model, spurious correlations may be easily discovered and leveraged by these review-based methods leading to higher attentions towards non-generalized features and achieving unexpected results.

4.2.2 Robustness Analysis (RQ2). To answer RQ2, we evaluate if the extracted rationales R are effective for recommendation across data distributions or even across datasets. The intuition is from d-separation in Figure 1, in which C_* and Y are independent conditioned on R. Thus, models based on R should be less influenced when C_* is changed, e.g., shifting data distribution or testing on other similar dataset.

To evaluate the performance of R3 with data distribution shifts, we conduct detailed experiments on the debiased test sets namely test-u. As mentioned above, test-u contains reviews posted by users whose average rating is around 3, i.e., the test rating distributions are significantly different from the training rating distributions in this experiment. As shown in Table 5 (appendix), test-u shows a more uniform distribution among ratings from 1 to 5. Table 1 (b) compares the accuracy of R3 and all the baseline methods on the four debiased test sets, in which we can see that R3 outperforms all the other methods on all the datasets. We can further have the following observations: 1) the rationales extracted by R3 can better help to alleviate the spurious correlation effects than the other methods and thus can help to achieve more robust results on datasets with new rating distributions and 2) our rationale predictor R3-R performs better than the correlation predictor R3-C in this setting because R3-C takes not only rationales but also correlations as its input, which may introduce unexpected spurious correlations even though rationales are with higher weights in the model.

To evaluate the performance of R3 across different datasets, we train R3 and the state-of-the-art baseline NARRE on the Health-and-Personal-Care dataset and evaluate their performance on the Beauty dataset. Since these are two different datasets, MF may suffer from untrained item embeddings, so that we only test the review-based methods. We choose NARRE as the baseline here because it shows very competitive results in previous experiments and it also follows the classic $\textbf{interest} + \textbf{bias}_{\textbf{user}} + \textbf{bias}_{\textbf{item}} + \textbf{bias}_{\textbf{global}}$ prediction paradigm so that we can easily analyse the effectiveness of each component. The components are defined as follows:

$$\begin{aligned} \pi_{u,v} &= \text{interest} + \text{bias}_{\text{user}} + \text{bias}_{\text{item}} + \text{bias}_{\text{global}}. \\ \pi &= \text{interest}. \\ \pi_u &= \text{interest} + \text{bias}_{\text{user}}. \\ \pi_v &= \text{interest} + \text{bias}_{\text{item}}. \end{aligned} \tag{17}$$

Here, **interest** means that we predict the ratings by user embedding and item embedding via dot production or MLP.

Intuitively, in this transfer learning setting, all predictions involving item features or item bias will be non-transferable because the items in test set were not observable during training. Therefore, user preferences will be the only transferable features in this experiment. As shown in Table 2, both R3-R and R3-C exhibit higher accuracy than NARRE on the new dataset, which indicates that our methods can better capture the invariant features across datasets than NARRE. In addition, we can have the following two observations: 1) we can see from the Pearson Correlation Coefficient (PCC) results that R3 relies more on π_u (the only transferable feature) in the prediction than NARRE; and 2) π and π_v are closely related to item features and item biases, which are non-transferable across datasets, and their correlations with R3 are close to zero. These

[3]https://nlp.stanford.edu/projects/glove/

Table 2: Transfer learning from Health-and-Personal-Care to Beauty, i.e., we train R3 and NARRE on Health-and-Personal-Care and evaluate their performance on Beauty. The best performance is in boldface. *: p < 0.05 in statistical significance test, for R3-R compared to NARRE.

Model	MSE	PCC			
		$\pi_{u,v}$	π	π_u	π_v
NARRE	1.4687	0.1446	0.0840	0.1446	0.0840
R3-R	1.4267*	0.1638	-0.0328	0.1638	-0.0328
R3-C	**1.4156**	0.1639	-0.0022	0.1639	-0.0022

two observations further confirm that R3 can better capture the transferable features when making predictions across datasets.

Note that R3-C achieves slightly better accuracy than R3-R in the transfer learning setting. The reason is that only user preferences and user biases are transferable in this setting but π will be non-transferable because it makes predictions by both user embedding (transferable) and item embedding (non-transferable). Thus, the only transferable feature is the user bias term. Since R3-C can better capture training data biases than R3-R, R3-C may exhibit higher accuracy when the predictions are only relying on the user biases.

4.2.3 Impact of Each Component (RQ3). Since R3 follows the classic $\textbf{interest} + \textbf{bias}_{\textbf{user}} + \textbf{bias}_{\textbf{item}} + \textbf{bias}_{\textbf{global}}$ prediction paradigm, we measure the impact of each component by PCC. Intuitively, higher PCC of a component indicates higher importance of its predictions. Table 3 presents the detailed MSE and PCC comparison among R3, MF and NARRE on the Yelp dataset under two settings: benchmark test set (with the same rating distribution as the training set) and debiased test set (with different rating distribution as the training set).

From the results in the benchmark setting, we can have the following observations: 1) both the PCCs of π_u and π_v are significant, i.e., bias terms contribute a lot to the predictions. The item bias dominates because π_v has the highest PCC value; 2) R3-R achieves higher accuracy but lower PCC on user/item bias compared to MF and NARRE, which indicates that R3-R can achieve decent performance even without fully relying on rating distribution biases; 3) R3-C can also achieve decent performance without fully relying on rating distribution biases, which should be due to that rationale features will be given higher weights in R3-C so that R3-C can rely on these rationale features in addition to user/item biases.

From the results in the debiased setting, we can have the following observations: 1) the users are new in this setting, so that π and π_u will be uncorrelated to the predictions in MF. But review-based methods will not suffer from new user issue, because they can rely on reviews in rating prediction; 2) in R3-R, the PCCs of π_u and π_v are almost equal, which indicates that the rationale features captured by R3-R are predictive even on new users with new rating distributions; and 3) in R3-C, the PCC of π_v is much higher than that of π_u, which indicates that the features extracted by R3-C are biased. Note that the MSE difference between R3-R and R3-C is not statistically significant and should be due to the randomness in the test set.

Table 3: Detailed MSE and PCC comparison among R3, MF and NARRE on the Yelp dataset under two settings. The best performance is in boldface. **: p < 0.01 in statistical significance test, for R3-R compared to the best baseline.

Model	MSE	PCC			
		$\pi_{u,v}$	π	π_u	π_v
benchmark test set					
MF	1.0811	0.3731	0.0470	0.2017	0.3325
NARRE	1.0672	0.3671	0.2810	0.2307	0.3490
R3-R	**1.0603****	0.3696	0.1799	0.1965	0.3290
R3-C	1.0638	0.3677	0.1638	0.2027	0.3280
debiased test set					
MF	2.6131	0.3257	-0.0004	-0.0004	0.3257
NARRE	2.8603	0.3237	0.2603	0.2603	0.3237
R3-R	2.3466**	0.3290	0.3128	0.3128	0.3287
R3-C	**2.3432**	0.3323	0.2209	0.2209	0.3275

4.2.4 Explanation via Rationales (RQ4). In this experiment, we analyze the explainability of the proposed method. State-of-the-art explainability metrics are based on calculating the proportion of explainable items among all recommendations. Inspired by [1], we use Explainability Recall to measure explainability. Original explainability recall is the proportion of explainable items in the top-n recommendation list relative to the number of all explainable items for a given user. However, all of the recommendations provided by the review based recommender systems can be explained using reviews used for recommendations. Hence we propose to fine tune the original explainability recall for finer grained explanation fidelity evaluation, we calculate the proportion of raionales extracted by our models relative to the rationales provided by the user in its ground truth review, which is unseen to our model. More specifically, we first compare the rationales extracted by the proposed rationale generator with the aspects and opinions extracted by Snippext [25] in each review. Intuitively, aspects and opinions in each review are causal features for predicting the ratings since they are the most descriptive features in reviews. And Snippext is the state-of-the-art aspects and opinions extractor. Hence, a high explainability recall between the rationales and the aspects and opinions indicates that the proposed rationale generator can extract more of the true rationales. Since it is common that the extracted rationales and aspects and opinion tokens are with the same semantic meaning but represented by different words (e.g., nice and good), so we compute the cosine similarity between their word vectors and treat cosine similarities above a certain threshold as true positives.

Figure 3 shows the correlation between cosine similarity threshold and the explainability recall. The intuitive understanding behind the threshold is as follows: if we select a cosine similarity threshold as 0.7, it will infer that words like "aged" and "older" share the same semantic. As we can see from the results, if we select the threshold as 0.7, we can achieve a pretty high explainability recall of 0.9, i.e., around 90% of aspects and opinions are successfully discovered by the proposed rationale generator.

Figure 3 also compares the explainability of R3 with the state-of-the-art explainable method – AR [27]. As we can see, AR shows a model fidelity of 0.85 in this experiment, which indicates that 85% of

Table 4: Rationales (words in boldface) from user and item reviews extracted by R3 for predicting the targeted review and rating. The targeted review is: food has improved recently but service is slow and always pesky flies - gross. its hit or miss.

user historical reviews	item historical reviews
- been here a few times - close location and "burger deals". **service** is **horrible** and **food** is so **slow**, could be better. cute burger ideas, but not worth full price - more **bar** like then food place. - water as beverage - **horrid service** - to get water then food, oh boy then waiting for bill. **food** was **good** , but **service blew**!	have been there multiple times as 1/2 off coupons - **bad food** and **service** 3 out of 4 visits - 1/2 off coupons still available - no thank u. even when empty **service sucked**, **burgers overcooked**, side **dish errors** on plate, bad bad, was great place - bar seems **busy** for happy hour maybe, no more attempts by me.

the recommendations are explainable. Same as explainability recall, model fidelity also measures the percentage of explainable recommendations. This comparison confirms that R3 exhibits comparable explainability to the state-of-the-art explainable CF method.

Besides quantitative analysis, we further present some rationale examples extracted by the proposed method in Table 4. Consider the context of recommending a restaurant to a user. R3 can find that the user was more concerned with food quality and service quality, which were reflected by his/her reviews about food and service. Meanwhile, R3 also finds the restaurant reviews that contain aspects and opinions about food and service. Then, R3 can make recommendations based on the rationales from user historical reviews and item historical reviews. The targeted review (the ground truth of the example) verifies that the extracted rationales from historical reviews are indeed highly consistent with the targeted review. This qualitative study further confirms the effectiveness of the proposed rationale generator.

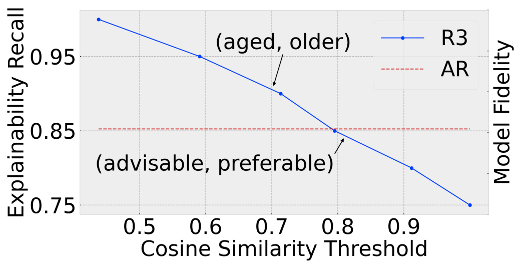

Figure 3: Explainability Recall of R3 w.r.t. different word vector cosine similarity thresholds (x-axis). Model fidelity of the AR method is also reported for comparison.

5 RELATED WORK

Many recent works are proposed to exploit reviews for recommendation, which mainly falls into two categories. 1) Topic-based methods [2, 24, 37, 47], which integrates topic models in their frameworks to learn the latent factors from user and item reviews to improve the recommendation performance. However, many of these methods organize reviews via the bag-of-words representation, which ignores the word ordering and may fail to effectively capture the semantic meanings of reviews. 2) Deep learning-based methods [5, 8, 38, 48], which try to extract features from user reviews and item reviews through deep learning architectures, and use these extracted features to perform matrix factorization. Recently, [14] introduced novel views of organizing and exploiting reviews: sequence level and graph level, which enables the CF models to additionally capture short-term features and collaborative features. Different from the above methods, this work proposes a novel method to extract rationales from reviews to achieve more robust recommendation.

Selective rationalization [6, 9, 10, 20, 45] is also a related area, which can help to address the interpretability issue. The key idea behind these methods is to find a small subset of the input features – rationales – that suffice on its own to yield the same outcome. Lei et al. [20] first proposed a generator-predictor framework for rationalization, following which several recent works [6, 9, 10, 45] applied different techniques to deal with the challenges in generating high-quality rationales. Recently, Chang et al. [7] proposed a game-theoretic invariant rationalization criterion which can learn invariant rationales among different environments. This work follows this line of research and sheds new light on applying rationalization techniques for accurate and explainable recommendation.

It has been widely known that virtually all data for training CF models suffers from selection bias, because the data we observe is missing not at random [23, 33–35, 44]. For instance, users are less likely to provide ratings for movies they do not like [23]. Previous approaches for dealing with selection bias include choosing more robust target metrics [35] or modeling the generative process of missing data explicitly [16]. Recently, several approaches [33, 34, 44] took a counterfactual perspective, resulting in a principled approach for adjusting for selection bias via propensities. This work differs from the above works by taking a SCM perspective and proposes a review rationalization method to achieve accurate and explainable recommendation, which can alleviate the spurious correlations caused by the missing not at random issue.

6 CONCLUSION

In this paper, we propose a review rationalization based algorithm for accurate and explainable recommendation, named Recommendation via Review Rationalization (R3). R3 first extracts rationales from user and item reviews via a rationale generator to alleviate the effects of spurious correlations in recommendation. Then, R3 can achieve accurate recommendation and provide causal-aware explanation based on the rationales. Extensive experiments on public datasets demonstrate the superiority of the proposed method in terms of accuracy and explainability.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 62172106 and 61932007.

REFERENCES

[1] Behnoush Abdollahi and Olfa Nasraoui. 2017. Using explainability for constrained matrix factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 79–83.

[2] Yang Bao, Hui Fang, and Jie Zhang. 2014. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *Twenty-Eighth AAAI conference on artificial intelligence*.

[3] Alex Beutel, Ed H Chi, Zhiyuan Cheng, Hubert Pham, and John Anderson. 2017. Beyond globally optimal: Focused learning for improved recommendations. In *Proceedings of the 26th International Conference on World Wide Web*. 203–212.

[4] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. 891–900.

[5] Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 288–296.

[6] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2019. *A Game Theoretic Approach to Class-wise Selective Rationalization*. Vol. 32. 10055–10065.

[7] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2020. Invariant rationalization. In *International Conference on Machine Learning*. PMLR, 1448–1458.

[8] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*. 1583–1592.

[9] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*. PMLR, 883–892.

[10] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2018. L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data. (2018).

[11] Yu-Neng Chuang, Chih-Ming Chen, Chuan-Ju Wang, Ming-Feng Tsai, Yuan Fang, and Ee-Peng Lim. 2020. TPR: Text-aware Preference Ranking for Recommender Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 215–224.

[12] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research* 12, ARTICLE (2011), 2493–2537.

[13] Xin Dong, Jingchao Ni, Wei Cheng, Zhengzhang Chen, Bo Zong, Dongjin Song, Yanchi Liu, Haifeng Chen, and Gerard De Melo. 2020. Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation. 34, 05 (2020), 7667–7674.

[14] Jingyue Gao, Yang Lin, Yasha Wang, Xiting Wang, Zhao Yang, Yuanduo He, and Xu Chu. 2020. Set-Sequence-Graph: A Multi-View Approach Towards Exploiting Reviews for Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 395–404.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[16] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic Matrix Factorization with Non-Random Missing Data. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32* (Beijing, China) *(ICML'14)*. JMLR.org, II–1512–II–1520.

[17] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

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

[19] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2019. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature communications* 10, 1 (2019), 1–8.

[20] Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 107–117.

[21] Dongsheng Li, Haodong Liu, Chao Chen, Yingying Zhao, Stephen M. Chu, and Bo Yang. 2021. NeuSE: A Neural Snapshot Ensemble Method for Collaborative Filtering. *ACM Transactions on Knowledge Discovery from Data* 15, 6 (2021).

[22] Pamela J Ludford, Dan Cosley, Dan Frankowski, and Loren Terveen. 2004. Think different: increasing online community participation using uniqueness and group dissimilarity. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 631–638.

[23] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence* (Vancouver, BC, Canada) *(UAI'07)*. AUAI Press, Arlington, Virginia, USA, 267–275.

[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*. 165–172.

[25] Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. 2020. Snippext: Semi-Supervised Opinion Mining with Augmented Data. In *Proceedings of The Web Conference 2020*. Association for Computing Machinery, New York, NY, USA, 617–628. https://doi.org/10.1145/3366423.3380144

[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] Georgina Peake and Jun Wang. 2018. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2060–2069.

[28] Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc.

[29] Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics surveys* 3 (2009), 96–146.

[30] Reinald Adrian Pugoy and Hung-Yu Kao. 2021. Unsupervised Extractive Summarization-Based Representations for Accurate and Explainable Collaborative Filtering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2981–2990.

[31] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[32] Noveen Sachdeva and Julian McAuley. 2020. How Useful Are Reviews for Recommendation? A Critical Review and Potential Improvements. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 1845–1848. https://doi.org/10.1145/3397271.3401281

[33] Tobias Schnabel and Paul N. Bennett. 2020. Debiasing Item-to-Item Recommendations With Small Annotated Datasets. In *Fourteenth ACM Conference on Recommender Systems*. Association for Computing Machinery, New York, NY, USA, 73–81. https://doi.org/10.1145/3383313.3412265

[34] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. PMLR, 1670–1679.

[35] Harald Steck. 2010. Training and Testing of Recommender Systems on Data Missing Not at Random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, DC, USA) *(KDD '10)*. Association for Computing Machinery, New York, NY, USA, 713–722. https://doi.org/10.1145/1835804.1835895

[36] Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. 2020. OpinionDigest: A Simple Framework for Opinion Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5789–5798.

[37] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-boosted latent topics: Understanding users and items with ratings and reviews.. In *IJCAI*, Vol. 16. 2640–2646.

[38] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2309–2318.

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

[40] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[41] Jiafeng Xia, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. 2021. Incremental Graph Convolutional Network for Collaborative Filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. ACM, New York, NY, USA, 2170–2179.

[42] Doris Xin, Nicolas Mayoraz, Hubert Pham, Karthik Lakshmanan, and John R Anderson. 2017. Folding: Why good models sometimes make spurious recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 201–209.

[43] Shuyuan Xu, Yunqi Li, Shuchang Liu, Zuohui Fu, Yingqiang Ge, Xu Chen, and Yongfeng Zhang. 2021. Learning causal explanations for recommendation. 2911 (2021), 13–25.

[44] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-Not-at-Random Implicit Feedback *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 279–287. https://doi.org/10.1145/3240323.3240355

[45] Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 2019. Rethinking Cooperative Rationalization: Introspective Extraction and Complement Control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 4094–4103.

[46] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101.
[47] 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.* 83–92.
[48] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (Cambridge, United Kingdom) *(WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 425–434. https://doi.org/10.1145/3018661.3018665

A TEXT PROCESSOR

In this section, we detail the text processor we use. The text processor works as follows. In the first layer, a word embedding function maps each word in the review into a d dimensional vector, and then the given text will be transformed to an embedded matrix with fixed length L (padded with zero wherever necessary to tackle length variations). The word embedding model can be any pre-trained models like those trained on the GoogleNews corpus using word2vec [26], or on Wikipedia using GloVe [4].

Following the embedding layer, there is a convolutional layer consisting of m neurons, each of which is associated with a filter $K \in \mathbb{R}^{k\times d}$ which produces features by applying convolution operator on word vectors. Let $\mathbf{e} = \{e_1, \ldots, e_L\}$ be the embedding matrix corresponding to the input text with length l ($l \leq L$). The j^{th} neuron produces its features as follows:

$$c_j = \mathbf{ReLU}(\mathbf{e} * K_j + b_j), \tag{18}$$

where b_j is the bias, $*$ is the convolution operation and **ReLU** is the Rectified Linear Unit activation function.

Let $c_1, c_2, \ldots, c_L$ be the features produced by the j^{th} neuron using the sliding window with size k over the embedded text, then we treat them as word-level features. To capture features of each review, we use a max-pooling operation [12] to capture the most important features. The final output of the convolutional layer is the concatenation of the output from all m neurons as follows:

$$\begin{aligned} o_j &= \max(c_1, c_2, \ldots, c_L), \\ O &= [o_1, o_2, \ldots, o_m]. \end{aligned} \tag{19}$$

B RATING DISTRIBUTIONS

In this section, we report rating distributions of all datasets we used for training and testing in Table 5.

Table 5: Rating Distributions of All Datasets.

Dataset	Split	5	4	3	2	1
Home and Kitchen	train	0.6843	0.1886	0.0697	0.0305	0.0268
	valid	0.6826	0.1905	0.0699	0.0289	0.0281
	test	0.6855	0.1878	0.0713	0.0301	0.0253
	test-u	0.2851	0.2094	0.1628	0.1391	0.2037
Amazon Toys and Games	train	0.6637	0.2175	0.0800	0.0246	0.0143
	valid	0.6528	0.2228	0.0834	0.0250	0.0160
	test	0.6743	0.2105	0.0780	0.0241	0.0132
	test-u	0.2448	0.2705	0.2251	0.1318	0.1279
Health and Personal Care	train	0.6638	0.1958	0.0813	0.0337	0.0253
	valid	0.6584	0.1927	0.0851	0.0363	0.0275
	test	0.6752	0.1869	0.0797	0.0329	0.0253
	test-u	0.2592	0.2139	0.1909	0.1431	0.1929
Beauty	test	0.5770	0.2002	0.1121	0.0577	0.0530
Yelp	train	0.4436	0.3211	0.1336	0.0553	0.0465
	valid	0.4385	0.3263	0.1348	0.0531	0.0473
	test	0.4605	0.3094	0.1279	0.0559	0.0465
	test-u	0.1838	0.2322	0.2091	0.1690	0.2060

C RANKING METRIC

In this section, we evaluate the proposed method in top-N recommendation task and report the results on ranking metrics. Table 6 compares the Normalized Discounted Cumulative Gain (NDCG) of R3 with MF and NARRE.

Table 6: Detailed NDCG comparison among R3, MF and NARRE on the Yelp dataset under two settings. The best performance is in boldface. **: p < 0.01 in statistical significance test, for R3-R compared to the best baseline.

Metric	Model			
	MF	NARRE	R3-R	R3-C
benchmark test set				
NDCG	0.0975	0.1053	**0.1803****	0.1798
debiased test set				
NDCG	0.2072	0.2075	**0.2082****	0.2079