On the Relationship between Explanation and Recommendation: Learning to Rank Explanations for Improved Performance

LEI LI, Hong Kong Baptist University, China
YONGFENG ZHANG, Rutgers University, USA
LI CHEN, Hong Kong Baptist University, China

Explaining to users why some items are recommended is critical, as it can help users to make better decisions, increase their satisfaction, and gain their trust in recommender systems (RS). However, existing explainable RS usually consider explanation as a side output of the recommendation model, which has two problems: (1) It is difficult to evaluate the produced explanations, because they are usually model-dependent, and (2) as a result, how the explanations impact the recommendation performance is less investigated.

In this article, explaining recommendations is formulated as a ranking task and learned from data, similarly to item ranking for recommendation. This makes it possible for standard evaluation of explanations via ranking metrics (e.g., Normalized Discounted Cumulative Gain). Furthermore, this article extends traditional item ranking to an item–explanation joint-ranking formalization to study if purposely selecting explanations could reach certain learning goals, e.g., improving recommendation performance. A great challenge, however, is that the sparsity issue in the user-item-explanation data would be inevitably severer than that in traditional user–item interaction data, since not every user–item pair can be associated with all explanations. To mitigate this issue, this article proposes to perform two sets of matrix factorization by considering the ternary relationship as two groups of binary relationships. Experiments on three large datasets verify the solution's effectiveness on both explanation ranking and item recommendation.

CCS Concepts: • **Information systems** → **Recommender systems**; **Learning to rank**; • **Computing methodologies** → **Multi-task learning;**

Additional Key Words and Phrases: Explainable recommendation, explanation ranking, learning to explain

ACM Reference format:
Lei Li, Yongfeng Zhang, and Li Chen. 2023. On the Relationship between Explanation and Recommendation: Learning to Rank Explanations for Improved Performance. *ACM Trans. Intell. Syst. Technol.* 14, 2, Article 21 (February 2023), 24 pages.
https://doi.org/10.1145/3569423

This work was supported by Hong Kong RGC/GRF (RGC/HKBU12201620) and partially supported by NSF IIS-1910154, 2007907, and 2046457. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.
Authors' addresses: L. Li and L. Chen, Hong Kong Baptist University, 34 Renfrew Road, Hong Kong, 000000, China; emails: {csleili, lichen}@comp.hkbu.edu.hk; Y. Zhang, Rutgers University, 110 Frelinghuysen Road, New Brunswick, New Jersey, 08854, USA; email: yongfeng.zhang@rutgers.edu.

2157-6904/2023/02-ART21 $15.00
https://doi.org/10.1145/3569423

1 INTRODUCTION

Recommendation algorithms, such as collaborative filtering [38, 39] and matrix factorization [23, 34], have been widely deployed in online platforms, such as e-commerce and social networks, to help users find their interested items. Meanwhile, there is a growing interest in explainable recommendation [5, 9, 11, 12, 15, 17, 26, 29, 47, 52, 53], which aims at producing user-comprehensible explanations, as they can help users make informed decisions and gain users' trust in the system [42, 52]. However, in current explainable recommendation approaches, explanation is often a side output of the model, which would incur two problems: first, the standard evaluation of explainable recommendation could be difficult, because the explanations vary from model to model (i.e., model-dependent); second, these approaches rarely study the potential impacts of explanations, mainly because of the first problem.

Evaluation of explanations in existing works can be generally classified into four categories, including case study, user study, online evaluation, and offline evaluation [52]. In most works, case study is adopted to show how the example explanations are correlated with recommendations. These examples may look intuitive, but they are less representative to reflect the overall quality of the explanations. Results of user study [1, 16] are more plausible, but it can be expensive and is usually evaluated in simulated environments that may not reflect real users' actual perception. Though this is not a problem in online evaluation, it is difficult to implement as it relies on the collaboration with industrial firms, which may explain why only few works [33, 49, 53] conducted online evaluation. Consequently, one may wonder whether it is possible to evaluate the explainability using offline metrics. However, as far as we know, there is no standard metrics that are well recognized by the community. Though BLEU [35] and ROUGE [30] have been widely adopted to evaluate text quality for natural language generation, text quality is not equal to explainability [6, 26].

With the attempt to achieve a standard offline evaluation of recommendation explanations, we formulate the explanation problem as a ranking task [31]. The basic idea is to train a model that can select appropriate explanations from a shared explanation pool for a recommendation. For example, when a movie recommender system suggests the movie "Frozen" to a user, it may also provide a few explanations, such as "great family movie" and "excellent graphics," as shown in Figure 1. Notice that these explanations are available all the time, but their ranking orders differ for different movie recommendations, and only those ranked top are presented to the user. In this case, the explanations are also learned from data, similarly to recommendations. Moreover, this general formulation can be adapted to various explanation styles, such as sentences, images, and even new styles yet to be invented, as long as the user-item-explanation interactions are available. As an instantiation, we adopt three public datasets with textual explanations [27] for experimentation.

With the evaluation and data, we can investigate the potential impacts of explanations, such as higher chance of item click, conversion, or fairness [41], which are less explored but are particularly important in commercial systems. Without an appropriate approach to explanation evaluation, explanations have usually been modeled as an auxiliary function of the recommendation task in most explainable models [5, 11, 32, 40, 53]. Recent works that jointly model recommendation and text generation [12] or feature prediction [15, 45] find that the two tasks could influence each other. In particular, Reference [10] shows that fine-tuning the parallel task of feature ranking can boost the recommendation performance. Moreover, a user study shows that users' feedback on explanation items could help to improve recommendation accuracy [16]. Based on these findings, we design an item–explanation joint-ranking framework to study if showing some particular explanations could lead to increased item acceptance rate (i.e., improving the recommendation performance). Furthermore, we are motivated to identify how the recommendation task and the explanation task

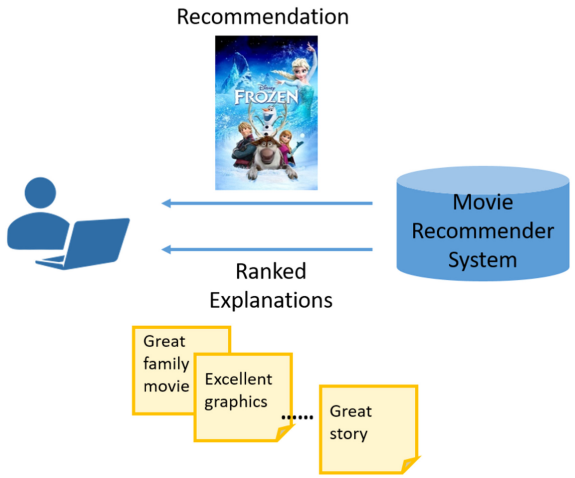


Fig. 1. A toy example of explanation ranking for a movie recommender system.

would interact with each other, whether there is a tradeoff between them, and how to achieve the most ideal solution for both.

However, the above investigation cannot proceed without addressing the inherent data sparsity issue in the user-item-explanation interactions. In traditional pairwise data, each user may be associated with several items, but in the user-item-explanation triplet data, each user–item pair may be associated with only one explanation. In consequence, the data sparsity problem is severer for explanation ranking. Therefore, how to design an effective model for such one-shot learning scenario becomes a great challenge. Our solution is to separate user-item-explanation triplets into user–explanation and item–explanation pairs, which significantly alleviates the data sparsity problem. Based on this idea, we design two types of model. First, a general model that only makes use of IDs, aims to accommodate a variety of explanation styles, such as sentences and images. Second, a domain-specific model based on BERT [14] further leverages the semantic features of the explanations to enhance the ranking performance.

In summary, our key contributions are as follows:

- To the best of our knowledge, our work is the first attempt to achieve standard evaluation of explainability for explainable recommendation via well-recognized metrics, such as **Normalized Discounted Cumulative Gain (NDCG)**, precision, and recall. We realize this by formulating the explanation problem as a ranking-oriented task.
- With the evaluation, we further propose an item–explanation joint-ranking framework that can reach our designed goal, i.e., improving the performance of both recommendation and explanation, as evidenced by our experimental results.
- To that end, we address the data sparsity issue in the explanation ranking task by designing an effective solution, being applied to two types of models (with and without semantic features of the explanations).[1] Extensive experiments show their effectiveness against strong baselines.

In the following, we first summarize related work in Section 2 and then formulate the problems in Section 3. Our proposed models and the joint-ranking framework are presented in Section 4.

[1]Codes available at https://github.com/lileipisces/BPER.

Section 5 introduces the experimental setup, and the discussion of results is provided in Section 6. We conclude this work with outlooks in Section 7.

2 RELATED WORK

Recent years have witnessed a growing interest in explainable recommendation [4, 5, 9, 11, 12, 15, 25, 26, 29, 32, 40, 47, 53]. In these works, there is a variety of explanation styles to recommendations, including visual highlights [9], textual highlights [32, 40], item neighbors [16], knowledge graph paths [7, 20, 48], word cloud [53], item features [17], pre-defined templates [15, 25, 53], automatically generated text [12, 26, 28, 29, 50, 51], retrieved text [4, 5, 11, 46, 47], and so on. The last type of style is related to this article, but explanations in these works are merely side outputs of their recommendation models. As a result, none of these works measured the explanation quality based on benchmark metrics. In comparison, we formulate the explanation task as a learning to rank [31] problem, which enables standard offline evaluation via ranking-oriented metrics.

On the one hand, the application of learning to rank can also be found in other domains. For instance, References [19, 44] attempt to explain entity relationships in Knowledge Graphs. The major difference from our work is that they heavily rely on the semantic features of explanations, either constructed manually [44] or extracted automatically [19], while one of our models works well when leveraging only the relation of explanations to users and items, without considering such features.

On the other hand, the appropriateness of current evaluation for explanations is still under debate. There are some works [12, 29] that regard text similarity metrics (i.e., BLEU [35] in machine translation and ROUGE [30] in text summarization) as explainability, when generating textual reviews/tips for recommendations. However, text similarity does not equal to explainability [6, 26]. For example, when the ground-truth is "*sushi is good*," two generated explanations, "*ramen is good*" and "*sushi is delicious*," gain the same score on the two metrics. However, from the perspective of explainability, the latter is obviously more related to the ground-truth, as they both refer to the same feature "*sushi*," but the metrics fail to reflect this issue. As a response, in this article we propose a new evaluation approach based on ranking.

Our proposed models are experimented on textual datasets, but it can be applied to a broad spectrum of other explanation styles, e.g., images, as discussed earlier. Concretely, on each dataset there is a pool of candidate explanations to be selected for each user–item pair. A recent online experiment [49] conducted on Microsoft Office 365[2] shows that these types of globally shared explanations are indeed helpful to users. The main focus of this work is to study how users perceive explanations, which is different from ours that aims to design effective models to rank explanations. Despite that, their research findings motivate us to provide better explanations that could lead to improved recommendations.

In more details, we model the user-item-explanation relations for both item and explanation ranking. There is a previous work [17] that similarly considers user-item-aspect relations as a tripartite graph, where aspects are extracted from user reviews. Another branch of related work is tag recommendation for folksonomy [22, 37], where tags are ranked for each given user–item pair. In terms of problem setting, our work is different from the preceding two, because they solely rank either items/aspects [17] or tags [22, 37], while besides that we also rank item–explanation pairs as a whole in our joint-ranking framework. Another difference is that we study how semantic features of explanations could help enhance the performance of explanation ranking, while none of them did so.

[2] https://www.office.com.

Table 1. Key Notations and Concepts

Symbol	Description
$\mathcal{T}$	training set
$\mathcal{U}$	set of users
$\mathcal{I}$	set of items
$\mathcal{I}_u$	set of items that user u preferred
$\mathcal{E}$	set of explanations
$\mathcal{E}_u$	set of user u's explanations
$\mathcal{E}_i$	set of item i's explanations
$\mathcal{E}_{u,i}$	set of explanations that user u preferred w.r.t. item i
$\mathbf{P}$	latent factor matrix for users
$\mathbf{Q}$	latent factor matrix for items
$\mathbf{O}$	latent factor matrix for explanations
$\mathbf{p}_u$	latent factors of user u
$\mathbf{q}_i$	latent factors of item i
$\mathbf{o}_e$	latent factors of explanation e
b_i	bias term of item i
b_e	bias term of explanation e
d	dimension of latent factors
α, λ	regularization coefficient
γ	learning rate
T	iteration number
M	number of recommendations for each user
N	number of explanations for each recommendation
$\hat{r}_{u,i}$	score predicted for user u on item i
$\hat{r}_{u,i,e}$	score predicted for user u on explanation e of item i

3 PROBLEM FORMULATION

The key notations and concepts for the problems are presented in Table 1. We use $\mathcal{U}$ to denote the set of all users, $\mathcal{I}$ the set of all items, and $\mathcal{E}$ the set of all explanations. Then the historical interaction set is given by $\mathcal{T} \subseteq \mathcal{U} \times \mathcal{I} \times \mathcal{E}$ (an illustrating example of such interaction is depicted in Figure 2). In the following, we first introduce item ranking and explanation ranking, respectively, and then the item–explanation joint-ranking.

3.1 Item Ranking

Personalized recommendation aims at providing a user with a ranked list of items that he/she never interacted with before. For each user $u \in \mathcal{U}$, the list of M items can be generated as follows:

$$\text{Top}(u, M) := \underset{i \in \mathcal{I} / \mathcal{I}_u}{\arg\max^{M}} \hat{r}_{u,\underline{i}}, \tag{1}$$

where $\hat{r}_{u,\underline{i}}$ is the predicted score for a user u on item i and $\mathcal{I} / \mathcal{I}_u$ denotes the set of items on which user u has no interactions. In Equation (1), i is underlined, which means that we aim to rank the items.

3.2 Explanation Ranking

Explanation ranking is the task of finding a list of appropriate explanations for a user–item pair to justify the recommendation. Formally, given a user $u \in \mathcal{U}$ and an item $i \in \mathcal{I}$, the goal of this

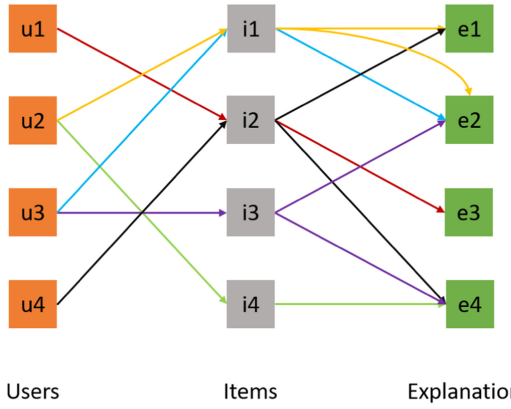


Fig. 2. Illustration of user-item-explanation interactions.

task is to rank the entire collection of explanations $\mathcal{E}$ and select the top N to reason why the item i is recommended. Specifically, we define this list of top N explanations as

$$\text{Top}(u, i, N) := \underset{e \in \mathcal{E}}{\arg\max^{N}} \hat{r}_{u,i,\underline{e}}, \tag{2}$$

where $\hat{r}_{u,i,\underline{e}}$ is the estimated score of explanation e for a given user–item pair (u, i), which could be given by a recommendation model or by the user's true behavior.

3.3 Item-Explanation Joint-Ranking

The preceding tasks solely rank either items or explanations. In this task, we further investigate whether it is possible to find an ideal item–explanation pair for a user and to whom the explanation best justifies the item that he/she likes the most. To this end, we treat each pair of item–explanation as a joint unit and then rank these units. Specifically, for each user $u \in \mathcal{U}$, a ranked list of M item–explanation pairs can be produced as follows:

$$\text{Top}(u, M) := \underset{i \in \mathcal{I}/\mathcal{I}_u, e \in \mathcal{E}}{\arg\max^{M}} \hat{r}_{u,\underline{i,e}}, \tag{3}$$

where $\hat{r}_{u,\underline{i,e}}$ is the predicted score for a given user u on the item–explanation pair (i, e).

We see that either item ranking task or explanation ranking task is a special case of this item–explanation joint-ranking task. Concretely, Equation (3) degenerates to Equation (1) when explanation e is fixed, while it reduces to Equation (2) if item i is already known.

4 OUR FRAMEWORK FOR RANKING TASKS

4.1 Joint-ranking Reformulation

Suppose we have an ideal model that can perform the aforementioned joint-ranking task. During the prediction stage as in Equation (3), there would be $|\mathcal{I}| \times |\mathcal{E}|$ candidate item–explanation pairs to rank for each user $u \in \mathcal{U}$. The runtime complexity is then $O(|\mathcal{U}| \cdot |\mathcal{I}| \cdot |\mathcal{E}|)$, which makes this task impractical compared with the traditional recommendation task's $O(|\mathcal{U}| \cdot |\mathcal{I}|)$ complexity.

To reduce the complexity, we reformulate the joint-ranking task by performing ranking for items and explanations simultaneously but separately. In this way, we are also able to investigate the relationship between item ranking and explanation ranking, e.g., improving the performance of both. Specifically, during the testing stage, we first follow Equation (1) to rank items for each user $u \in \mathcal{U}$, which has the runtime complexity of $O(|\mathcal{U}| \cdot |\mathcal{I}|)$. After that, for M recommendations for

each user, we can rank and select explanations to justify each of them according to Equation (2). The second step's complexity is $O(|\mathcal{U}| \cdot M \cdot |\mathcal{E}|)$, but since M is a constant and $|\mathcal{E}| \ll |\mathcal{I}|$ (see Table 2), the overall complexity of the two steps is $O(|\mathcal{U}| \cdot |\mathcal{I}|)$.

In the following, we first analyze the drawback of a conventional **Tensor Factorization (TF)** model when being applied to the explanation ranking problem and then introduce our solution, **Bayesian Personalized Explanation Ranking (BPER)**. Second, we show how to further enhance BPER by utilizing the semantic features of textual explanations (denoted as BPER+). Third, we illustrate their relation to two typical TF methods, **Canonical Decomposition (CD)** and **Pairwise Interaction Tensor Factorization (PITF)**. Last, we integrate the explanation ranking with item ranking into a multi-task learning framework as a joint-ranking task.

4.2 Bayesian Personalized Explanation Ranking

To perform explanation ranking, the score $\hat{r}_{u,i,e}$ on each explanation $e \in \mathcal{E}$ for a given user–item pair (u, i) must be estimated. As the user-item-explanation ternary relations $\mathcal{T} = \{(u, i, e)|u \in \mathcal{U}, i \in \mathcal{I}, e \in \mathcal{E}\}$ form an interaction cube, we are inspired to employ factorization models to predict this type of scores. There are a number of tensor factorization techniques [2, 21], such as **Tucker Decomposition (TD)** [43], CD [3], and High Order Singular Value Decomposition [13]. Intuitively, one would adopt CD because of its linear runtime complexity in terms of both training and prediction [37] and its close relation to **Matrix Factorization (MF)** [34], which has been extensively studied in recent years for item recommendation. Formally, according to CD, the score $\hat{r}_{u,i,e}$ of user u on item i's explanation e can be estimated by the sum over the element-wise multiplication of the user's latent factors $\mathbf{p}_u$, the item's $\mathbf{q}_i$, and the explanation's $\mathbf{o}_e$,

$$\hat{r}_{u,i,e} = (\mathbf{p}_u \odot \mathbf{q}_i)^\top \mathbf{o}_e = \sum_{k=1}^{d} p_{u,k} \cdot q_{i,k} \cdot o_{e,k}, \tag{4}$$

where $\odot$ denotes the element-wise multiplication of two vectors.

However, this method may not be effective enough due to the inherent sparsity problem of the ternary data as we discussed before. Since each user–item pair (u, i) in the training set $\mathcal{T}$ is unlikely to have interactions with many explanations in $\mathcal{E}$, the data sparsity problem for explanation ranking is more severe than that for item recommendation. Simply multiplying the three vectors would hurt the performance of explanation ranking, which is evidenced by our experimental results in Section 6.

To mitigate such an issue and to improve the effectiveness of explanation ranking, we propose to separately estimate the user u's preference score $\hat{r}_{u,e}$ on explanation e and the item i's appropriateness score $\hat{r}_{i,e}$ for explanation e. To this end, we perform two sets of matrix factorization rather than employing one single TF model. In this way, the sparsity problem would be considerably alleviated, since the data are reduced to two collections of binary relations, both of which are similar to the case of item recommendation discussed above. Last, the two scores $\hat{r}_{u,e}$ and $\hat{r}_{i,e}$ are combined linearly through a hyper-parameter μ. Specifically, the score of user u for item i on explanation e is predicted as follows:

$$\begin{cases} \hat{r}_{u,e} = \mathbf{p}_u^\top \mathbf{o}_e^U + b_e^U = \sum_{k=1}^{d} p_{u,k} \cdot o_{e,k}^U + b_e^U \\ \hat{r}_{i,e} = \mathbf{q}_i^\top \mathbf{o}_e^I + b_e^I = \sum_{k=1}^{d} q_{i,k} \cdot o_{e,k}^I + b_e^I \\ \hat{r}_{u,i,e} = \mu \cdot \hat{r}_{u,e} + (1 - \mu) \cdot \hat{r}_{i,e} \end{cases}, \tag{5}$$

where $\{\mathbf{o}_e^U, b_e^U\}$ and $\{\mathbf{o}_e^I, b_e^I\}$ are two different sets of latent factors for explanations, corresponding to users and items, respectively.

Since selecting explanations that are likely to be perceived helpful by users is inherently a ranking-oriented task, directly modeling the relative order of explanations is thus more effective than simply predicting their absolute scores. The **Bayesian Personalized Ranking (BPR)** criterion [36] meets such an optimization requirement. Intuitively, a user would be more likely to appreciate explanations that cater to her own preferences, while those that do not fit one's interests would be less attractive to the user. Similarly, some explanations might be more suitable to describe certain items, while other explanations might not. To build such types of pairwise preferences, we use the first two rows in Equation (5) to compute the difference between two explanations for both user u and item i as follows:

$$\begin{cases} \hat{r}_{u,ee'} = \hat{r}_{u,e} - \hat{r}_{u,e'} \\ \hat{r}_{i,ee''} = \hat{r}_{i,e} - \hat{r}_{i,e''} \end{cases}, \tag{6}$$

which respectively reflect user u's interest in explanation e over e' and item i's appropriateness for explanation e over e''.

With the scores $\hat{r}_{u,ee'}$ and $\hat{r}_{u,ee''}$, we can then adopt the BPR criterion [36] to minimize the following objective function:

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u} \sum_{e \in \mathcal{E}_{u,i}} \left[\sum_{e' \in \mathcal{E}/\mathcal{E}_u} -\ln \sigma(\hat{r}_{u,ee'}) + \sum_{e'' \in \mathcal{E}/\mathcal{E}_i} -\ln \sigma(\hat{r}_{i,ee''}) \right] + \lambda \, ||\Theta||_F^2 , \tag{7}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\mathcal{I}_u$ represents the set of items that user u has interacted with, $\mathcal{E}_{u,i}$ is the set of explanations in the training set for the user–item pair (u, i), $\mathcal{E}/\mathcal{E}_u$ and $\mathcal{E}/\mathcal{E}_i$ respectively correspond to explanations that user u and item i have not interacted with, Θ is the model parameter, and λ is the regularization coefficient.

From Equation (7), we can see that there are two explanation tasks to be learned respectively, corresponding to users and items. During the training stage, we allow them to be equally important, since we have a hyper-parameter μ in Equation (5) to balance their importance during the testing stage. The effect of this parameter is studied in Section 6.1. After the model parameters are estimated, we can rank explanations according to Equation (2) for each user–item pair in the testing set. As we model the explanation ranking task under BPR criterion, we accordingly name our method BPER. To learn the model parameter Θ, we draw on the widely used stochastic gradient descent algorithm to optimize the objective function in Equation (7). Specifically, we first randomly initialize the parameters and then repeatedly update them by uniformly taking samples from the training set and computing the gradients w.r.t. the parameters, until the convergence of the algorithm. The complete learning steps are shown in Algorithm 1.

4.3 BERT-enhanced BPER (BPER+)

The BPER model only exploits the IDs of users, items, and explanations to infer their relation for explanation ranking. However, this makes the rich semantic features of the explanations, which could also capture the relation between explanations, under-explored. For example, "*the acting is good*" and "*the acting is great*" for movie recommendation both convey a positive sentiment with a similar meaning, so their ranks are expected to be close. Hence, we further investigate whether such features could help to enhance BPER. As a feature extractor, we opt for BERT [14], a well-known pre-trained language model, whose effectiveness has been demonstrated on a wide range of natural language understanding tasks. Specifically, we first add a special [CLS] token at the beginning of a textual explanation e, e.g., "*[CLS] the acting is great.*" After passing it through BERT, we can obtain the aggregate representation (corresponding to [CLS]) that encodes the explanation's overall semantics. To match the dimension of latent factors in our model, we apply a linear layer to this vector, resulting in $\mathbf{o}_e^{BERT}$. Then, we enhance the two ID-based explanation vectors $\mathbf{o}_e^U$ and

ALGORITHM 1: Bayesian Personalized Explanation Ranking

Input: training set $\mathcal{T}$, dimension of latent factors d, learning rate γ, regularization coefficient λ, iteration number T

Output: model parameters $\Theta = \{\mathbf{P}, \mathbf{Q}, \mathbf{O}^U, \mathbf{O}^I, \mathbf{b}^U, \mathbf{b}^I\}$

Initialize Θ, including $\mathbf{P} \leftarrow \mathbb{R}^{|\mathcal{U}|\times d}$, $\mathbf{Q} \leftarrow \mathbb{R}^{|\mathcal{I}|\times d}$, $\mathbf{O}^U \leftarrow \mathbb{R}^{|\mathcal{E}|\times d}$, $\mathbf{O}^I \leftarrow \mathbb{R}^{|\mathcal{E}|\times d}$, $\mathbf{b}^U \leftarrow \mathbb{R}^{|\mathcal{E}|}$, $\mathbf{b}^I \leftarrow \mathbb{R}^{|\mathcal{E}|}$
for $t_1 = 1$ to T **do**
  for $t_2 = 1$ to $|\mathcal{T}|$ **do**
    Uniformly draw (u, i, e) from $\mathcal{T}$, e' from $\mathcal{E}/\mathcal{E}_u$, and e'' from $\mathcal{E}/\mathcal{E}_i$
    $\hat{r}_{u,ee'} \leftarrow \hat{r}_{u,e} - \hat{r}_{u,e'}$, $\hat{r}_{i,ee''} \leftarrow \hat{r}_{i,e} - \hat{r}_{i,e''}$
    $x \leftarrow -\sigma(-\hat{r}_{u,ee'})$, $y \leftarrow -\sigma(-\hat{r}_{i,ee''})$
    $\mathbf{p}_u \leftarrow \mathbf{p}_u - \gamma \cdot (x \cdot (\mathbf{o}_e^U - \mathbf{o}_{e'}^U) + \lambda \cdot \mathbf{p}_u)$
    $\mathbf{q}_i \leftarrow \mathbf{q}_i - \gamma \cdot (y \cdot (\mathbf{o}_e^I - \mathbf{o}_{e''}^I) + \lambda \cdot \mathbf{q}_i)$
    $\mathbf{o}_e^U \leftarrow \mathbf{o}_e^U - \gamma \cdot (x \cdot \mathbf{p}_u + \lambda \cdot \mathbf{o}_e^U)$
    $\mathbf{o}_{e'}^U \leftarrow \mathbf{o}_{e'}^U - \gamma \cdot (-x \cdot \mathbf{p}_u + \lambda \cdot \mathbf{o}_{e'}^U)$
    $\mathbf{o}_e^I \leftarrow \mathbf{o}_e^I - \gamma \cdot (y \cdot \mathbf{q}_i + \lambda \cdot \mathbf{o}_e^I)$
    $\mathbf{o}_{e''}^I \leftarrow \mathbf{o}_{e''}^I - \gamma \cdot (-y \cdot \mathbf{q}_i + \lambda \cdot \mathbf{o}_{e''}^I)$
    $b_e^U \leftarrow b_e^U - \gamma \cdot (x + \lambda \cdot b_e^U)$
    $b_{e'}^U \leftarrow b_{e'}^U - \gamma \cdot (-x + \lambda \cdot b_{e'}^U)$
    $b_e^I \leftarrow b_e^I - \gamma \cdot (y + \lambda \cdot b_e^I)$
    $b_{e''}^I \leftarrow b_{e''}^I - \gamma \cdot (-y + \lambda \cdot b_{e''}^I)$
  end for
end for

$\mathbf{o}_e^I$ in Equation (5) by multiplying $\mathbf{o}_e^{BERT}$, resulting in $\mathbf{o}_e^{U+}$ and $\mathbf{o}_e^{I+}$,

$$\begin{cases} \mathbf{o}_e^{U+} = \mathbf{o}_e^U \odot \mathbf{o}_e^{BERT} \\ \mathbf{o}_e^{I+} = \mathbf{o}_e^I \odot \mathbf{o}_e^{BERT} \end{cases} . \tag{8}$$

To predict the score for the (u, i, e) triplet, we replace $\mathbf{o}_e^U$ and $\mathbf{o}_e^I$ in Equation (5) with $\mathbf{o}_e^{U+}$ and $\mathbf{o}_e^{I+}$. Then we use Equation (7) as the objective function, which can be optimized via back-propagation. In Equation (8), we adopt the multiplication operation simply to verify the feasibility of incorporating semantic features. The model may be further improved by more sophisticated operations, e.g., multi-layer perceptron, but we leave the exploration for future work.

Notice that BPER is a general method that only requires the IDs of users, items, and explanations, which makes it very flexible when being adapted to other explanation styles (e.g., images [9]). However, it may suffer from the common cold-start issue as with other recommender systems. BPER+ could mitigate this issue to some extent, because besides IDs it also considers the semantic relation between textual explanations via BERT, which can connect new explanations with existing ones. As the first work on ranking explanations for recommendations, we opt to make both methods relatively simple for reproducibility purpose. In this way, it is also easy to observe the experimental results (such as the impact of explanation task on recommendation task), without the interference of other factors.

4.4 Relation among BPER, BPER+, CD, and PITF

In fact, our BPER model is a type of TF, so we analyze its relation to two closely related TF methods: CD [3] and PITF [37]. On the one hand, in theory BPER can be considered as a special case of the CD model. Suppose the dimensionality of BPER is $2 \cdot d + 2$. We can reformulate it as CD in the

following:

$$
\begin{aligned}
p_{u,k}^{CD} &= \begin{cases} \mu \cdot p_{u,k}, & \text{if } k \le d \\ \mu, & \text{else} \end{cases}, \\
q_{i,k}^{CD} &= \begin{cases} (1-\mu) \cdot q_{i,k}, & \text{if } k > d \text{ and } k \le 2 \cdot d \\ 1-\mu, & \text{else} \end{cases}, \\
o_{e,k}^{CD} &= \begin{cases} o_{e,k}^{U}, & \text{if } k \le d \\ o_{e,k}^{I}, & \text{else if } k \le 2 \cdot d \\ b_e^U, & \text{else if } k = 2 \cdot d + 1 \\ b_e^I, & \text{else} \end{cases},
\end{aligned}
\tag{9}
$$

where the parameter μ is a constant.

On the other hand, PITF can be seen as a special case of our BPER. Formally, its predicted score $\hat{r}_{u,i,e}$ for the user-item-explanation triplet (u, i, e) can be calculated by

$$
\hat{r}_{u,i,e} = \mathbf{p}_u^\top \mathbf{o}_e^U + \mathbf{q}_i^\top \mathbf{o}_e^I = \sum_{k=1}^{d} p_{u,k} \cdot o_{e,k}^U + \sum_{k=1}^{d} q_{i,k} \cdot o_{e,k}^I. \tag{10}
$$

We can see that our BPER degenerates to PITF if in Equation (5) we remove the bias terms b_e^U and b_e^I and set the hyper-parameter μ to 0.5, which means that the two types of scores for users and items are equally important to the explanation ranking task.

Although CD is more general than our BPER, its performance may be affected by the data sparsity issue as discussed before. Our BPER could mitigate this problem given its explicitly designed structure that may be difficult for CD to learn from scratch. When comparing with PITF, we can find that the parameter μ in BPER is able to balance the importance of the two types of scores, corresponding to users and items, which makes our BPER more expressive than PITF and hence likely reach better ranking quality.

In a similar way, BPER+ can also be rewritten as CD or PITF. Concretely, by revising the last part of Equation (9) as the following formula, BPER+ can be seen as CD. When $\mathbf{o}_e^{BERT} = [1, \ldots, 1]^\top$, BPER+ is equal to BPER, so it can be easily converted into PITF. The graphical illustration of the four models is shown in Figure 3,

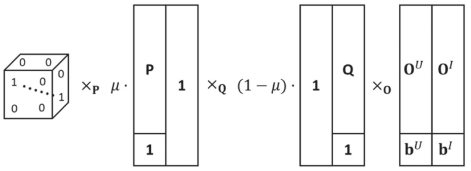
$$
o_{e,k}^{CD} = \begin{cases} o_{e,k}^{U} \cdot o_{e,k}^{BERT}, & \text{if } k \le d \\ o_{e,k}^{I} \cdot o_{e,k}^{BERT}, & \text{else if } k \le 2 \cdot d \\ b_e^U, & \text{else if } k = 2 \cdot d + 1 \\ b_e^I, & \text{else} \end{cases}. \tag{11}
$$

4.5 Joint-Ranking on BPER (BPER-J)

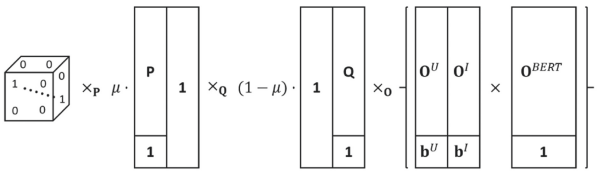
Owing to BPER's flexibility to accommodate various explanation styles as discussed before, we perform the joint-ranking on it. Specifically, we incorporate the two tasks of explanation ranking and item recommendation into a unified multi-task learning framework so as to find a good solution that benefits both of them.

For recommendation, we adopt the Singular Value Decomposition model [23] to predict the score $\hat{r}_{u,i}$ of user u on item i:

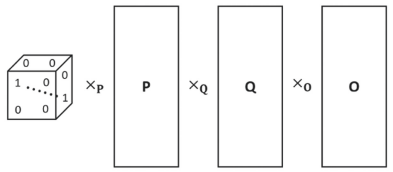
$$
\hat{r}_{u,i} = \mathbf{p}_u^\top \mathbf{q}_i + b_i = \sum_{k=1}^{d} p_{u,k} \cdot q_{i,k} + b_i, \tag{12}
$$



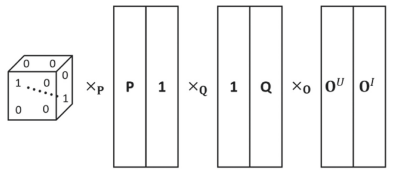
(a) Our Bayesian Personalized Explanation Ranking (BPER)



(b) Our BERT-enhanced BPER (BPER+)



(c) Canonical Decomposition (CD)



(d) Pairwise Interaction Tensor Factorization (PITF)

Fig. 3. Tensor Factorization models. The three matrices (i.e., **P**, **Q**, **O**) are model parameters. Our BPER and BPER+ can be regarded as special cases of CD, while PITF can be seen as a special case of our BPER and BPER+.

where b_i is the bias term for item i. Notice that the latent factors $\mathbf{p}_u$ and $\mathbf{q}_i$ are shared with those for explanation ranking in Equation (5). In essence, item recommendation is also a ranking task that can be optimized using BPR criteria [36], so we first compute the preference difference $\hat{r}_{u,ii'}$ between a pair of items i and i' to a user u as follows:

$$\hat{r}_{u,ii'} = \hat{r}_{u,i} - \hat{r}_{u,i'}, \tag{13}$$

which can then be combined with the task of explanation ranking in Equation (7) to form the following objective function for joint-ranking:

$$\begin{aligned} \min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u} \Bigg[\sum_{i' \in \mathcal{I} / \mathcal{I}_u} - \ln \sigma(\hat{r}_{u,ii'}) + \alpha \sum_{e \in \mathcal{E}_{u,i}} \Bigg(\sum_{e' \in \mathcal{E} / \mathcal{E}_u} \\ - \ln \sigma(\hat{r}_{u,ee'}) + \sum_{e'' \in \mathcal{E} / \mathcal{E}_i} - \ln \sigma(\hat{r}_{i,ee''}) \Bigg) \Bigg] + \lambda \, ||\Theta||_F^2 , \end{aligned} \tag{14}$$

where the parameter α can be fine-tuned to balance the learning of the two tasks.

We name this method BPER-J where J denotes joint-ranking. Similarly to BPER, we can update each parameter of BPER-J via stochastic gradient descent (see Algorithm 2).

5 EXPERIMENTAL SETUP

5.1 Datasets

To compare the ranking performance of different methods, it is expected that the datasets contain user-item-explanation interaction triplets. The datasets could be manually constructed as in Reference [49], but we are not given access to such datasets. Therefore, we adopt three public datasets[3] [27], where the explanations are automatically extracted from user reviews via near-duplicate detection, which ensures that the explanations are commonly used by users. Specifically,

[3] https://github.com/lileipisces/EXTRA.

ALGORITHM 2: Joint-Ranking on BPER (BPER-J)

Input: training set $\mathcal{T}$, dimension of latent factors d, learning rate γ, regularization coefficients α and λ, iteration number T

Output: model parameters $\Theta = \{\mathbf{P}, \mathbf{Q}, \mathbf{O}^U, \mathbf{O}^I, \mathbf{b}, \mathbf{b}^U, \mathbf{b}^I\}$

Initialize Θ, including $\mathbf{P} \leftarrow \mathbb{R}^{|\mathcal{U}|\times d}$, $\mathbf{Q} \leftarrow \mathbb{R}^{|\mathcal{I}|\times d}$, $\mathbf{O}^U \leftarrow \mathbb{R}^{|\mathcal{E}|\times d}$, $\mathbf{O}^I \leftarrow \mathbb{R}^{|\mathcal{E}|\times d}$, $\mathbf{b} \leftarrow \mathbb{R}^{|\mathcal{I}|}$, $\mathbf{b}^U \leftarrow \mathbb{R}^{|\mathcal{E}|}$, $\mathbf{b}^I \leftarrow \mathbb{R}^{|\mathcal{E}|}$
for $t_1 = 1$ to T **do**
  for $t_2 = 1$ to $|\mathcal{T}|$ **do**
    Uniformly draw (u, i, e) from $\mathcal{T}$, e' from $\mathcal{E}/\mathcal{E}_u$, e'' from $\mathcal{E}/\mathcal{E}_i$, and i' from $\mathcal{I}/\mathcal{I}_u$
    $\hat{r}_{u,ee'} \leftarrow \hat{r}_{u,e} - \hat{r}_{u,e'}$, $\hat{r}_{i,ee''} \leftarrow \hat{r}_{i,e} - \hat{r}_{i,e''}$, $\hat{r}_{u,ii'} \leftarrow \hat{r}_{u,i} - \hat{r}_{u,i'}$
    $x \leftarrow -\alpha \cdot \sigma(-\hat{r}_{u,ee'})$, $y \leftarrow -\alpha \cdot \sigma(-\hat{r}_{i,ee''})$, $z \leftarrow -\sigma(-\hat{r}_{u,ii'})$
    $\mathbf{p}_u \leftarrow \mathbf{p}_u - \gamma \cdot (x \cdot (\mathbf{o}_e^U - \mathbf{o}_{e'}^U) + z \cdot (\mathbf{q}_i - \mathbf{q}_{i'}) + \lambda \cdot \mathbf{p}_u)$
    $\mathbf{q}_i \leftarrow \mathbf{q}_i - \gamma \cdot (y \cdot (\mathbf{o}_e^I - \mathbf{o}_{e''}^I) + z \cdot \mathbf{p}_u + \lambda \cdot \mathbf{q}_i)$
    $\mathbf{q}_{i'} \leftarrow \mathbf{q}_{i'} - \gamma \cdot (-z \cdot \mathbf{p}_u + \lambda \cdot \mathbf{q}_{i'})$
    $\mathbf{o}_e^U \leftarrow \mathbf{o}_e^U - \gamma \cdot (x \cdot \mathbf{p}_u + \lambda \cdot \mathbf{o}_e^U)$
    $\mathbf{o}_{e'}^U \leftarrow \mathbf{o}_{e'}^U - \gamma \cdot (-x \cdot \mathbf{p}_u + \lambda \cdot \mathbf{o}_{e'}^U)$
    $\mathbf{o}_e^I \leftarrow \mathbf{o}_e^I - \gamma \cdot (y \cdot \mathbf{q}_i + \lambda \cdot \mathbf{o}_e^I)$
    $\mathbf{o}_{e''}^I \leftarrow \mathbf{o}_{e''}^I - \gamma \cdot (-y \cdot \mathbf{q}_i + \lambda \cdot \mathbf{o}_{e''}^I)$
    $b_i \leftarrow b_i - \gamma \cdot (z + \lambda \cdot b_i)$
    $b_{i'} \leftarrow b_{i'} - \gamma \cdot (-z + \lambda \cdot b_{i'})$
    $b_e^U \leftarrow b_e^U - \gamma \cdot (x + \lambda \cdot b_e^U)$
    $b_{e'}^U \leftarrow b_{e'}^U - \gamma \cdot (-x + \lambda \cdot b_{e'}^U)$
    $b_e^I \leftarrow b_e^I - \gamma \cdot (y + \lambda \cdot b_e^I)$
    $b_{e''}^I \leftarrow b_{e''}^I - \gamma \cdot (-y + \lambda \cdot b_{e''}^I)$
  end for
end for

the datasets are from different domains, including Amazon Movies & TV,[4] TripAdvisor[5] for hotels, and Yelp[6] for restaurants. Each record in the three datasets consists of user ID, item ID, and one or multiple explanation IDs and thus results in one or multiple user-item-explanation triplets. Moreover, each explanation ID appears no less than 5 times. The statistics of the three datasets are presented in Table 2. As it can be seen, the data sparsity issue on the three datasets is very severe.

Table 3 shows five example explanations taken from the three datasets. As we can see, all the explanations are quite concise and informative, which could prevent from overwhelming users, a critical issue for explainable recommendation [18]. Also, short explanations can be mobile-friendly, since it is difficult for a small screen to fit much content. Moreover, the explanations from different datasets well suit the target application domains, such as "*a wonderful movie for all ages*" for movies and "*comfortable hotel with good facilities*" for hotels. Explanations with negative sentiment can also be observed, e.g., "*the place is awful*," which can be used to justify why some items are dis-recommended [53]. Hence, we believe that the datasets are very suitable for our explanation ranking experiment.

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

Table 2. Statistics of the Datasets

	Amazon Movies & TV	TripAdvisor	Yelp
# of users	109,121	123,374	895,729
# of items	47,113	200,475	164,779
# of explanations	33,767	76,293	126,696
# of (u, i) pairs	569,838	1,377,605	2,608,860
# of (u, i, e) triplets	793,481	2,618,340	3,875,118
# of explanations/(u, i) pair	1.39	1.90	1.49
Density ($\times 10^{-10}$)	45.71	13.88	2.07

Density is #triplets divided by #users × #items × #explanations.

Table 3. Example Explanations on the Three Datasets

Amazon Movies & TV
Great story
Don't waste your money
The acting is great
The sound is okay
A wonderful movie for all ages
TripAdvisor
Great location
The room was clean
The staff were friendly and helpful
Bad service
Comfortable hotel with good facilities
Yelp
Great service
Everything was delicious
Prices are reasonable
This place is awful
The place was clean and the food was good

5.2 Compared Methods

To evaluate the performance of explanation ranking task, where the user–item pairs are given, we adopt the following baselines. Notice that we omit the comparison with TD [43], because it takes cubic time to run and we also find that it does not perform better than CD in our trial experiment.

- **RAND:** This is a weak baseline that randomly picks up explanations from the explanation collection $\mathcal{E}$. It is devised to examine whether personalization is needed for explanation ranking.
- **Revised User-based Collaborative Filtering (RUCF):** Because traditional CF methods [38, 39] cannot be directly applied to the ternary data, we make some modifications to their formula, following Reference [22]. The similarity between two users is measured by their associated explanation sets via Jaccard Index. When predicting a score for the (u, i, e) triplet, we first find users associated with the same item i and explanation e, i.e., $\mathcal{U}_i \cap \mathcal{U}_e$, from

which we then find the ones appearing in user u's neighbor set $\mathcal{N}_u$,

$$\hat{r}_{u,i,e} = \sum_{u' \in \mathcal{N}_u \cap (\mathcal{U}_i \cap \mathcal{U}_e)} s_{u,u'} \text{ where } s_{u,u'} = \frac{|\mathcal{E}_u \cap \mathcal{E}_{u'}|}{|\mathcal{E}_u \cup \mathcal{E}_{u'}|}. \tag{15}$$

- **Revised Item-based Collaborative Filtering (RICF):** This method predicts a score for a triplet from the perspective of items, whose formula is similar to Equation (15).
- **CD** [3] as shown in Equation (4): This method only predicts one score instead of two for the triplet (u, i, e), so its objective function shown below is slightly different from ours in Equation (7),

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u} \sum_{e \in \mathcal{E}_{u,i}} \sum_{e' \in \mathcal{E} / \mathcal{E}_{u,i}} -\ln \sigma(\hat{r}_{u,i,ee'}) + \lambda \, ||\Theta||_F^2 , \tag{16}$$

  where $\hat{r}_{u,i,ee'} = \hat{r}_{u,i,e} - \hat{r}_{u,i,e'}$ is the score difference between a pair of interactions.
- **PITF** [37]: This makes prediction for a triplet based on Equation (10), and its objective function is identical to CD's in Equation (16).

To verify the effectiveness of the joint-ranking framework, in addition to our method BPER-J, we also present the results of two baselines: CD [3] and PITF [37]. Since CD and PITF are not originally designed to accomplish the two tasks of item recommendation and explanation ranking together, we first allow them to make prediction for a user–item pair (u, i) via the inner product of their latent factors, i.e., $\hat{r}_{u,i} = \mathbf{p}_u^T \mathbf{q}_i$, and then combine this task with explanation ranking in a multi-task learning framework whose objective function is given as follows:

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u} \left[\sum_{i' \in \mathcal{I} / \mathcal{I}_u} -\ln \sigma(\hat{r}_{u,ii'}) + \alpha \sum_{e \in \mathcal{E}_{u,i}} \sum_{e' \in \mathcal{E} / \mathcal{E}_{u,i}} -\ln \sigma(\hat{r}_{u,i,ee'}) \right] + \lambda \, ||\Theta||_F^2 , \tag{17}$$

where $\hat{r}_{u,ii'} = \hat{r}_{u,i} - \hat{r}_{u,i'}$ is the difference between a pair of records. We name them CD-J and PITF-J, respectively, where J denotes joint-ranking.

5.3 Evaluation Metrics

To evaluate the performance of both recommendation and explanation, we adopt four commonly used ranking-oriented metrics in recommender systems: NDCG, Precision, Recall, and F1. We evaluate on top-10 ranking for both recommendation and explanation tasks. For the former task, it is easy to find the definition of the metrics in previous works, so we define those for the latter. Specifically, the scores for a user–item pair on the four metrics are computed as follows:

$$\begin{aligned}
\text{rel}_p &= \delta(\text{Top}(u, i, N, p) \in \mathcal{E}_{u,i}^{te}) \\
\text{NDCG}(u, i, N) &= \frac{1}{Z} \sum_{p=1}^{N} \frac{2^{\text{rel}_p} - 1}{\log(p+1)}, \text{ where } Z = \sum_{p=1}^{N} \frac{1}{\log(p+1)} \\
\text{Precision}(u, i, N) &= \frac{1}{N} \sum_{p=1}^{N} \text{rel}_p \text{ and Recall}(u, i, N) = \frac{1}{\left|\mathcal{E}_{u,i}^{te}\right|} \sum_{p=1}^{N} \text{rel}_p \\
\text{F1}(u, i, N) &= 2 \times \frac{\text{Precision}(u, i, N) \times \text{Recall}(u, i, N)}{\text{Precision}(u, i, N) + \text{Recall}(u, i, N)},
\end{aligned} \tag{18}$$

where rel_p indicates whether the pth explanation in the ranked list $\text{Top}(u, i, N)$ can be found in the ground-truth explanation set $\mathcal{E}_{u,i}^{te}$. Then, we can average the scores for all user–item pairs in the testing set.

5.4 Implementation Details

We randomly divide each dataset into training (70%) and testing (30%) sets and guarantee that each user/item/explanation has at least one record in the training set. The splitting process is repeated for 5 times. For validation, we randomly draw 10% records from training set. After hyper-parameters tuning, the average performance on the five testing sets is reported.

We implemented all the methods in Python.[7] For TF-based methods, including CD, PITF, CD-J, PITF-J, and our BPER and BPER-J, we search the dimension of latent factors d from [10, 20, 30, 40, 50], regularization coefficient λ from [0.001, 0.01, 0.1], learning rate γ from [0.001, 0.01, 0.1], and maximum iteration number T from [100, 500, 1000]. As to joint-ranking of CD-J, PITF-J and our BPER-J, the regularization coefficient α on explanation task is searched from [0, 0.1, ..., 0.9, 1]. For the evaluation of joint-ranking, we first evaluate the performance of item recommendation for users, followed by the evaluation of explanation ranking on those correctly predicted user–item pairs. For our methods BPER and BPER-J, the parameter μ that balances user and item scores for explanation ranking is searched from [0, 0.1, ..., 0.9, 1]. After parameter tuning, we use $d = 20$, $\lambda = 0.01$, $\gamma = 0.01$ and $T = 500$ for our methods, while the other parameters α and μ are dependent on the datasets.

The configuration of BPER+ is slightly different, because of the textual content of the explanations. We adopted the pre-trained BERT from huggingface,[8] and implemented the model in Python with PyTorch.[9] We set batch size to 128, $d = 20$, and $T = 5$. After parameter tuning, we set learning rate γ to 0.0001 on Amazon, and 0.00001 on both TripAdvisor and Yelp.

6 RESULTS AND ANALYSIS

In this section, we first compare our methods BPER and BPER+ with baselines regarding explanation ranking. Then, we study the capability of our methods in dealing with varying data sparseness. Third, we show a case study of explanation ranking for both recommendation and disrecommendation, and also present a small user study. Last, we analyze the joint-ranking results of three TF-based methods.

6.1 Comparison of Explanation Ranking

Experimental results for explanation ranking on the three datasets are shown in Table 4. We see that each method's performance on the four metrics (i.e., NDCG, Precision, Recall, and F1) are fairly consistent across the three datasets. The method RAND is among the weakest baselines, because it randomly selects explanations without considering user and item information, which implies that the explanation ranking task is non-trivial. CD performs even worse than RAND, because of the sparsity issue in the ternary data (see Table 2), for which CD may not be able to mitigate as discussed in Section 4.2. CF-based methods, i.e., RUCF and RICF, largely advance the performance of RAND, as they take into account the information of either users or items, which confirms the important role of personalization for explanation ranking. However, their performance is still limited due to data sparsity. PITF and our BPER/BPER+ outperform the CF-based methods by a large margin, as they not only address the data sparsity issue via their MF-like model structure but also take each user's and item's information into account using latent factors. Most importantly, our method BPER significantly outperforms the strongest baseline PITF, owing to its ability of producing two sets of scores, corresponding to users and items respectively, and its parameter μ that can balance their relative importance to explanation ranking. Last, BPER+ further improves BPER on

[7] https://www.python.org.
[8] https://huggingface.co/bert-base-uncased.
[9] https://pytorch.org.

Table 4. Performance Comparison of all Methods on the Top-10 Explanation Ranking in Terms of NDCG, Precision, Recall, and F1 (%)

	NDCG@10 (%)	Precision@10 (%)	Recall@10 (%)	F1@10 (%)	Training Time
	Amazon Movies & TV				
CD	0.001	0.001	0.007	0.002	1h48min
RAND	0.004	0.004	0.027	0.006	—
RUCF	0.341	0.170	1.455	0.301	—
RICF	0.417	0.259	1.797	0.433	—
PITF	2.352	1.824	14.125	3.149	1h51min
BPER	2.630*	**1.942***	**15.147***	**3.360***	1h56min
BPER+	**2.877***	1.919*	14.936*	3.317*	—
Improvement (%)	22.352	5.229	5.739	5.343	—
	TripAdvisor				
CD	0.001	0.001	0.003	0.001	5h32min
RAND	0.002	0.002	0.011	0.004	—
RUCF	0.260	0.151	0.779	0.242	—
RICF	0.031	0.020	0.087	0.030	—
PITF	1.239	1.111	5.851	1.788	7h9min
BPER	1.389*	1.236*	6.549*	1.992*	9h43min
BPER+	**2.096***	**1.565***	**8.151***	**2.515***	—
Improvement (%)	69.073	40.862	39.314	40.665	—
	Yelp				
CD	0.000	0.000	0.003	0.001	12h7min
RAND	0.001	0.001	0.007	0.002	—
RUCF	0.040	0.020	0.125	0.033	—
RICF	0.037	0.026	0.137	0.042	—
PITF	0.712	0.635	4.172	1.068	11h27min
BPER	0.814*	0.723*	**4.768***	1.218*	16h30min
BPER+	**0.903***	**0.731***	4.544*	**1.220***	—
Improvement (%)	26.861	15.230	8.925	14.228	—

The best performing values are in bold, and the second best underlined. Improvements are made by BPER+ over the best baseline PITF (* indicates the statistical significance over PITF for $p < 0.01$ via Student's t-test).

most of the metrics across the three datasets, especially on NDCG that cares about the ranking order, which can be attributed to the consideration of the semantic features of the explanations as well as BERT's strong language modeling capability to extract them.

Besides the explanation ranking performance, we also present the training time comparison of the three TF-based methods in Table 4. For fair comparison, the runtime testing is conducted on the same research machine without GPU, because these methods are all implemented in pure Python without involving deep learning framework. From the table, we can see that the training time of the three methods is generally consistent on different datasets. CD takes the least time to train, PITF needs a bit more training time, while the duration of training our BPER is the longest. This is quite expected, since the model complexity grows larger from CD to PITF and BPER. However, the slightly sacrificed training time of BPER is quite acceptable, because the gap of training duration between the three methods is not very large, e.g., 5h32min for CD, 7h9min for PITF and 9h43min for BPER on TripAdvisor dataset.

Last, we further analyze the parameter μ of BPER that controls the contributions of user scores and item scores in Equation (5). As it can be seen in Figure 4, the curves of NDCG, Pre, Rec,

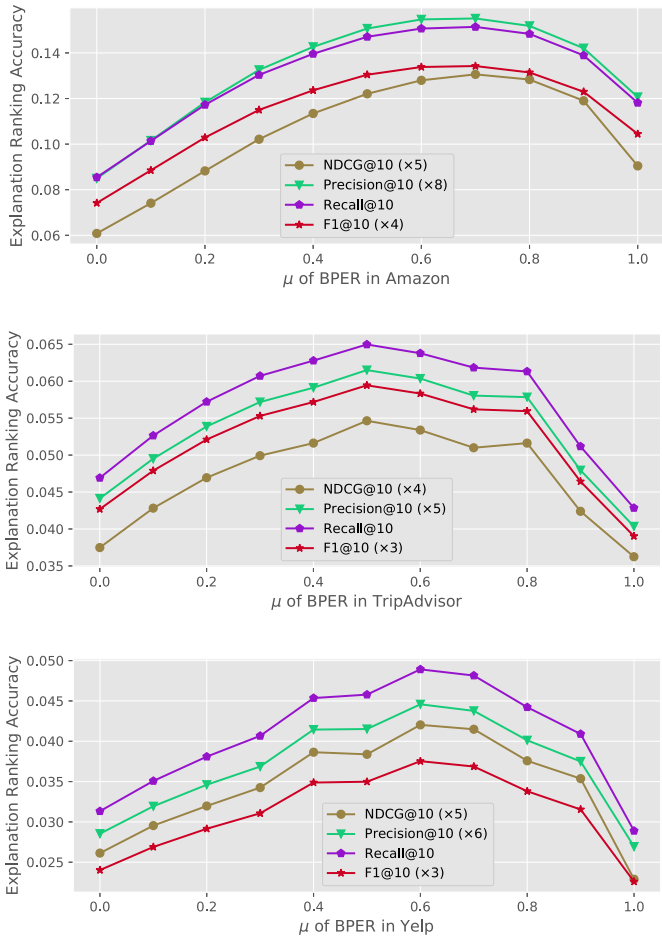


Fig. 4. The effect of μ in BPER on explanation ranking in three datasets. NDCG@10, Precision@10, and F1@10 are linearly scaled for better visualization.

and F1 are all bell shaped, where the performance improves significantly with the increase of μ until it reaches an optimal point, and then it drops sharply. Due to the characteristics of different application domains, the optimal points vary among the three datasets, i.e., 0.7 for both Amazon and Yelp and 0.5 for TripAdvisor. We omit the figures of BPER+, because the pattern is similar.

6.2 Results on Varying Data Sparseness

As discussed earlier, the sparsity issue of user-item-explanation triplewise data is more severe than that of traditional user–item pairwise data. To investigate how different methods deal with varying spareness, we further remove certain ratio of the Amazon training set, so that the training triplets to the whole dataset ranges from 30% to 70%, while the testing set remains untouched. For comparison with our BPER and BPER+, we include the most competitive baseline PITF. Figure 5 shows the ranking performance of the three methods w.r.t. varying spareness. The ranking results are quite consistent on the four metrics (i.e., NDCG, Precision, Recall, and F1). Moreover, with the increase of the amount of training triplets, the performance of all three methods goes up linearly. Particularly, the performance gap between our BPER/BPER+ and PITF is quite large, especially

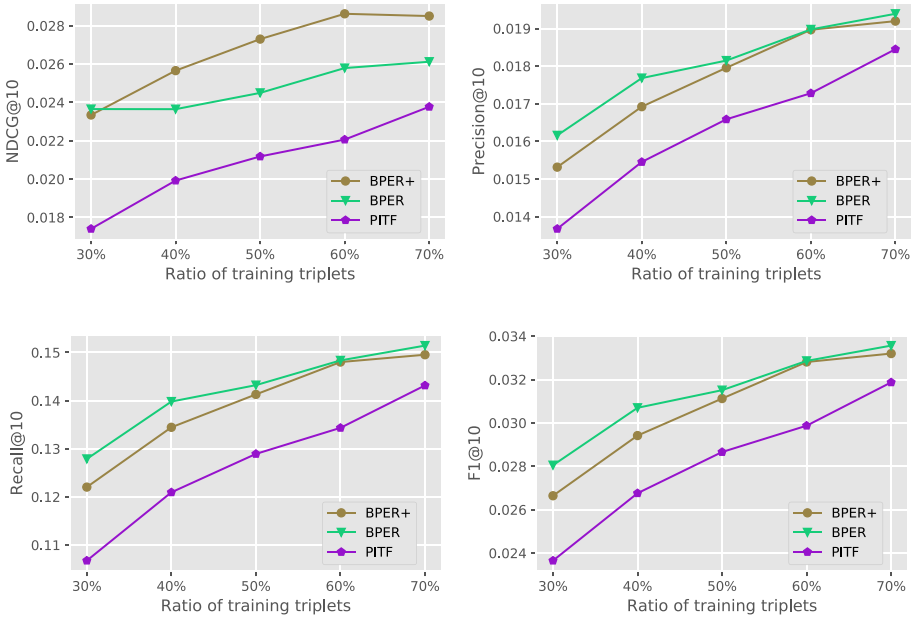


Fig. 5. Ranking performance of three TF-based methods w.r.t. varying sparseness of training data on Amazon dataset.

Table 5. Top-5 Explanations Selected by BPER and PITF for Two Given User–Item Pairs, Corresponding to Recommendation and Disrecommendation, on Amazon Movies & TV Dataset

Ground-truth	**BPER**	**PITF**
Special effects	*Special effects*	Great special effects
Great story	Good acting	Great visuals
Wonderful movie	This is a great movie	Great effects
	Great story	*Special effects*
	Great special effects	Good movie
The acting is terrible	*The acting is terrible*	Good action movie
	The acting is bad	Low budget
	The acting was horrible	Nothing special
	It's not funny	The acting is poor
	Bad dialogue	The acting is bad

The ground-truth explanations are unordered. Matched explanations are emphasized in italic font.

when the ratio of training data is small (e.g., 30%). These observations demonstrate our methods' better capability in mitigating data sparsity issue, and hence prove the rationale of our solution that converts triplets to two groups of binary relation.

6.3 Qualitative Case Study and User Study

To better understand how explanation ranking works, we first present a case study comparing our method BPER and the most effective baseline PITF on Amazon Movies & TV dataset in Table 5.

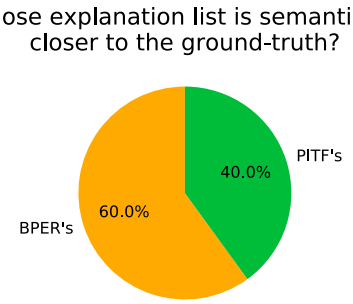


Fig. 6. Result of user study on explanations returned by two methods on Amazon Movies & TV dataset.

The two cases in the table respectively correspond to recommendation and disrecommendation. In the first case (i.e., recommendation), there are three ground-truth explanations, praising the movie's "*special effects*," "*story*," and overall quality. Generally speaking, the top-5 explanations resulting from both BPER and PITF are positive, and relevant to the ground-truth, because the two methods are both effective in terms of explanation ranking. However, since PITF's ranking ability is relatively weaker than our BPER, its explanations miss the key feature "*story*" that the user also cares about.

In the second case (i.e., disrecommendation), the ground-truth explanation is a negative comment about the target movie's "*acting*." Although the top explanations made by both BPER and PITF contain negative opinions regarding this aspect, their ranking positions are quite different (i.e., top-3 for our BPER vs. bottom-2 for PITF). Moreover, we notice that for this disrecommendation, PITF places a positive explanation in the first position, i.e., "*good action movie*," which not only contradicts the other two explanations, i.e., "*the acting is poor/bad*," but also mismatches the disrecommendation goal. Again, this showcases our model's effectiveness for explanation ranking.

We further conduct a small scale user study to investigate real people's perception toward the top ranked explanations. Specifically, we still compare our BPER with PITF on Amazon Movies & TV dataset. We prepared 10 different cases and hired college students to do the evaluation. In each case, we provide the movie's title and the ground-truth explanations, and ask the participants to select one explanation list that is semantically closer to the ground-truth. There are two randomly shuffled options returned by BPER and PITF, respectively. A case is valid only when at least two participants select the same option. The evaluation results are shown in Figure 6. We can see that on 60% of cases our BPER's explanations are closer to the ground-truth than PITF's, which is quite consistent with their explanation ranking performance.

6.4 Effect of Joint-Ranking

We perform joint-ranking for three TF-based models, i.e., BPER-J, CD-J, and PITF-J. Because of the consistency in the experimental results on different datasets, we only show results on Amazon and TripAdvisor. In Figure 7, we study the effect of the parameter α to both explanation ranking and item ranking in terms of F1 (results on the other three metrics are consistent). In each sub-figure, the green dotted line represents the performance of explanation ranking task without joint-ranking, whose value is taken from Table 4. As we can see, all the points on the explanation curve (in red) are above this line when α is greater than 0, suggesting that the explanation task benefits from the recommendation task under the joint-ranking framework. In particular, the explanation performance of CD-J improves dramatically under the joint-ranking framework, since

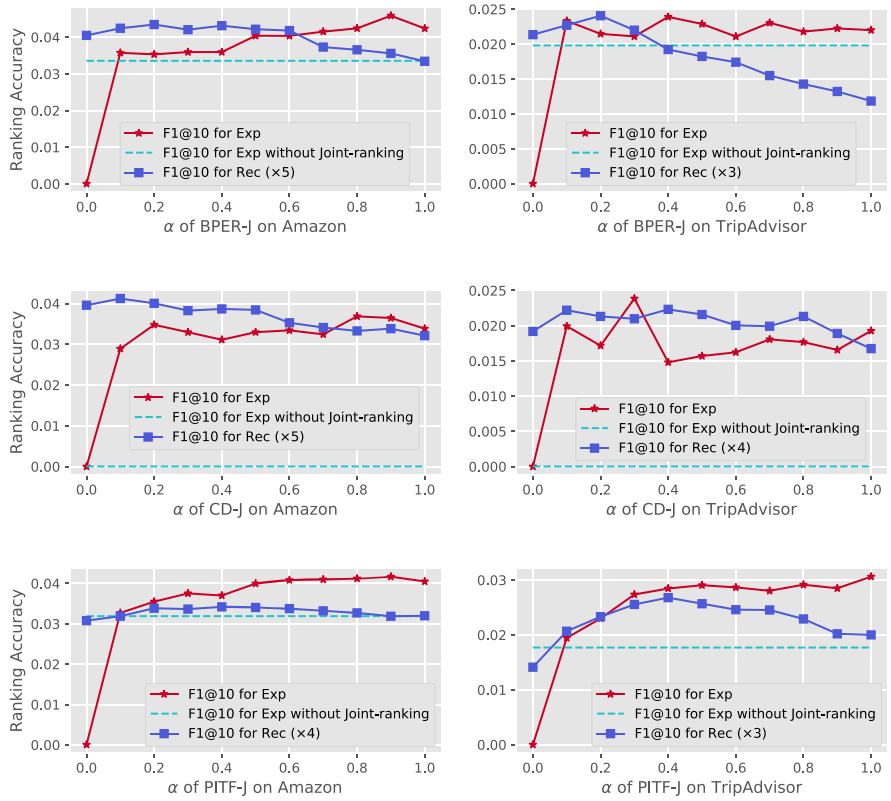


Fig. 7. The effect of α in three TF-based methods with joint-ranking on two datasets. Exp and Rec respectively denote the Explanation and Recommendation tasks. F1@10 for Rec is linearly scaled for better visualization.

its recommendation task suffers less from the data sparsity issue than the explanation task as discussed in Section 4.2. It in turn helps to better rank the explanations. Meanwhile, for the recommendation task, all the three models degenerate to BPR when α is set to 0. Therefore, on the recommendation curves (in blue), any points, whose values are greater than that of the starting point, gain profits from the explanation task as well. All these observations show the effectiveness of our joint-ranking framework in terms of enabling the two tasks to benefit from each other.

In Table 6, we make a self-comparison of the three methods in terms of NDCG and F1 (the other two metrics are similar). In this table, "Non-joint-ranking" corresponds to each model's performance with regard to explanation or recommendation when the two tasks are individually learned. In other words, the explanation performance is taken from Table 4, and the recommendation performance is evaluated when $\alpha = 0$. "Best Exp" and "Best Rec" denote the best performance of each method on respectively explanation task and recommendation task under the joint-ranking framework. As we can see, when the recommendation performance is the best for all the models with joint-ranking, the explanation performance is always improved. Although minor recommendation accuracy is sacrificed when the explanation task reaches the best performance, we can always find points where both of the two tasks are improved, e.g., on the top left of Figure 7 when α is in the range of 0.1 to 0.6 for BPER-J on Amazon. This again demonstrates our joint-ranking framework's capability in finding good solutions for both tasks.

Table 6. Self-comparison of Three TF-based Methods on Two Datasets with and without Joint-ranking in Terms of NDCG and F1

		Amazon				TripAdvisor			
		Exp (%)		Rec (‰)		Exp (%)		Rec (‰)	
		NDCG	F1	NDCG	F1	NDCG	F1	NDCG	F1
		BPER-J							
Non-joint-ranking		2.6	3.4	6.6	8.1	1.4	2.0	5.3	7.1
Joint-ranking	Best Exp	3.3 ↑	4.6 ↑	5.7 ↓	7.1 ↓	1.6 ↑	2.4 ↑	5.0 ↓	6.4 ↓
	Best Rec	2.6 ↕	3.5 ↑	7.1 ↑	8.7 ↑	1.5 ↑	2.1 ↑	6.3 ↑	8.0 ↑
Improvement (%)		26.9	35.3	7.6	7.4	14.3	20.0	18.9	11.3
		CD-J							
Non-joint-ranking		0.0	0.0	6.5	7.9	0.0	0.0	4.5	4.8
Joint-ranking	Best Exp	2.6 ↑	3.7 ↑	5.5 ↓	6.7 ↓	1.7 ↑	2.4 ↑	4.6 ↑	5.2 ↑
	Best Rec	1.9 ↑	2.9 ↑	6.8 ↑	8.2 ↑	9.6 ↑	1.5 ↑	4.9 ↑	5.6 ↑
Improvement (%)		Inf	Inf	4.6	3.8	Inf	Inf	8.9	16.7
		PITF-J							
Non-joint-ranking		2.4	3.2	6.5	7.7	1.2	1.8	4.3	4.7
Joint-ranking	Best Exp	3.0 ↑	4.2 ↑	6.4 ↓	8.0 ↑	2.0 ↑	2.9 ↑	6.0 ↑	7.6 ↑
	Best Rec	2.8 ↑	3.7 ↑	7.1 ↑	8.5 ↑	2.0 ↑	2.8 ↑	7.0 ↑	8.9 ↑
Improvement (%)		25.0	31.3	9.2	10.4	66.7	61.1	62.8	89.4

Top-10 results are evaluated for both explanation (Exp) and recommendation (Rec) tasks. The improvements are made by the best performance of each task under joint-ranking over that without it (i.e., in this case the two tasks are separately learned).

7 CONCLUSION AND FUTURE WORK

To the best of our knowledge, we are the first one that leverages standard offline metrics to evaluate explainability for explainable recommendation. We achieve this goal by formulating the explanation problem as a ranking task. With this quantitative measure of explainability, we design an item–explanation joint-ranking framework that can improve the performance of both recommendation and explanation tasks. To enable such joint-ranking, we develop two effective models to address the data sparsity issue, which were tested on three large datasets.

As future work, we are interested in considering the relationship (such as coherency [24] and diversity) between suggested explanations to further improve the explainability. In addition, we plan to conduct experiments in real-world systems to validate whether recommendations and their associated explanations as produced by the joint-ranking framework could influence users' behavior, e.g., clicking and purchasing. In addition, the joint-ranking framework in this article aims to improve the recommendation performance by providing explanations, while in the future, we will also consider improving other objectives based on explanations, such as recommendation serendipity [8] and fairness [41].

REFERENCES

[1] Krisztian Balog and Filip Radlinski. 2020. Measuring recommendation explanation quality: The conflicting goals of explanations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 329–338.

[2] Preeti Bhargava, Thomas Phan, Jiayu Zhou, and Juhan Lee. 2015. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *Proceedings of the 24th International Conference on World Wide Web*. 130–140.

[3] J. Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika* 35, 3 (1970), 283–319.
[4] Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 288–296.
[5] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the World Wide Web Conference*. 1583–1592.
[6] Hanxiong Chen, Xu Chen, Shaoyun Shi, and Yongfeng Zhang. 2019. Generate natural language explanations for recommendation. In *Proceedings of SIGIR'19 Workshop on ExplainAble Recommendation and Search*. ACM.
[7] Hongxu Chen, Yicong Li, Xiangguo Sun, Guandong Xu, and Hongzhi Yin. 2021. Temporal meta-path guided explainable recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 1056–1064.
[8] Li Chen, Yonghua Yang, Ningxia Wang, Keping Yang, and Quan Yuan. 2019. How serendipity improves user satisfaction with recommendations? A large-scale user evaluation. In *Proceedings of the World Wide Web Conference*. 240–250.
[9] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 765–774.
[10] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to rank features for recommendation over multiple categories. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 305–314.
[11] Xu Chen, Yongfeng Zhang, and Zheng Qin. 2019. Dynamic explainable recommendation based on neural attentive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 53–60.
[12] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. 2019. Co-attentive multi-task learning for explainable recommendation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'19)*. 2137–2143.
[13] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. 2000. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* 21, 4 (2000), 1253–1278.
[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*.
[15] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable recommendation through attentive multi-view learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3622–3629.
[16] Azin Ghazimatin, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. ELIXIR: Learning from user feedback on explanations to improve recommender models. In *Proceedings of the Web Conference 2021*. 3850–3860.
[17] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1661–1670.
[18] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. 241–250.
[19] Jizhou Huang, Wei Zhang, Shiqi Zhao, Shiqiang Ding, and Haifeng Wang. 2017. Learning to explain entity relationships by pairwise ranking with convolutional neural networks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'17)*. 4018–4025.
[20] Yafan Huang, Feng Zhao, Xiangyu Gui, and Hai Jin. 2021. Path-enhanced explainable recommendation with knowledge graphs. *World Wide Web* 24, 5 (2021), 1769–1789.
[21] Vassilis N. Ioannidis, Ahmed S. Zamzam, Georgios B. Giannakis, and Nicholas D. Sidiropoulos. 2019. Coupled graphs and tensor factorization for recommender systems and community detection. *IEEE Trans. Knowl. Data Eng.* 33, 3 (2019), 909–920.
[22] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. 2007. Tag recommendations in folksonomies. In *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 506–514.
[23] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[24] Trung-Hoang Le, Hady W. Lauw, and C. Bessiere. 2020. Synthesizing aspect-driven recommendation explanations from reviews. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IICAI'20)*. 2427–2434.
[25] Lei Li, Li Chen, and Ruihai Dong. 2021. CAESAR: Context-aware explanation based on supervised attention for service recommendations. *J. Intell. Inf. Syst.* 57, 1 (2021), 147–170.

[26] Lei Li, Yongfeng Zhang, and Li Chen. 2020. Generate neural template explanations for recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 755–764.
[27] Lei Li, Yongfeng Zhang, and Li Chen. 2021. EXTRA: Explanation ranking datasets for explainable recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
[28] Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.
[29] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 345–354.
[30] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. 74–81.
[31] Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval*. Springer Science & Business Media.
[32] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Coevolutionary recommendation model: Mutual learning between ratings and reviews. In *Proceedings of the World Wide Web Conference*. 773–782.
[33] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, exploit, and explain: Personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 31–39.
[34] Andriy Mnih and Russ R. Salakhutdinov. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*. 1257–1264.
[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. 311–318.
[36] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*.
[37] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*. 81–90.
[38] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. 175–186.
[39] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. 285–295.
[40] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 297–305.
[41] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2219–2228.
[42] Nava Tintarev and Judith Masthoff. 2015. Explaining recommendations: Design and evaluation. In *Recommender Systems Handbook* (2nd ed.), Bracha Shapira (Ed.). Springer, Chapter 10, 353–382.
[43] Ledyard R. Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31, 3 (1966), 279–311.
[44] Nikos Voskarides, Edgar Meij, Manos Tsagkias, Maarten De Rijke, and Wouter Weerkamp. 2015. Learning to explain entity relationships in knowledge graphs. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 564–574.
[45] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *Proceedings of the Special Interest Group on Information Retrieval (SIGIR'18)*. ACM, 165–174.
[46] Peng Wang, Renqin Cai, and Hongning Wang. 2022. Graph-based extractive explainer for recommendations. In *Proceedings of the ACM Web Conference 2022*. 2163–2171.
[47] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018. A reinforcement learning framework for explainable recommendation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*. IEEE, 587–596.
[48] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 285–294.
[49] Xuhai Xu, Ahmed Hassan Awadallah, Susan T. Dumais, Farheen Omar, Bogdan Popp, Robert Rounthwaite, and Farnaz Jahanbakhsh. 2020. Understanding user behavior for document recommendation. In *Proceedings of the Web Conference 2020*. 3012–3018.

[50] Aobo Yang, Nan Wang, Renqin Cai, Hongbo Deng, and Hongning Wang. 2022. Comparative explanations of recommendations. In *Proceedings of the ACM Web Conference 2022*. 3113–3123.
[51] Aobo Yang, Nan Wang, Hongbo Deng, and Hongning Wang. 2021. Explanation as a defense of recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 1029–1037.
[52] Yongfeng Zhang and Xu Chen. 2020. Explainable recommendation: A survey and new perspectives. *Found. Trends Inf. Retriev.* 14, 1 (2020), 1–101.
[53] 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.

Received 8 March 2022; revised 13 September 2022; accepted 4 October 2022