Fine-Tuning Large Language Model Based Explainable Recommendation with Explainable Quality Reward

Mengyuan Yang[1], **Mengying Zhu**[1*], **Yan Wang**[2], **Linxun Chen**[3], **Yilei Zhao**[1], **Xiuyuan Wang**[1], **Bing Han**[3], **Xiaolin Zheng**[1], **Jianwei Yin**[1]

[1] Zhejiang University, China
[2] School of Computing, Macqaurie University, Australia
[3] MYbank, Ant Group, China
{yangmy412, mengyingzhu, yilei_zhao, xiuyuanwang, xlzheng}@zju.edu.cn, yan.wang@mq.edu.au
{linxun.clx, hanbing.hanbing}@antgroup.com, zjuyjw@cs.zju.edu.cn

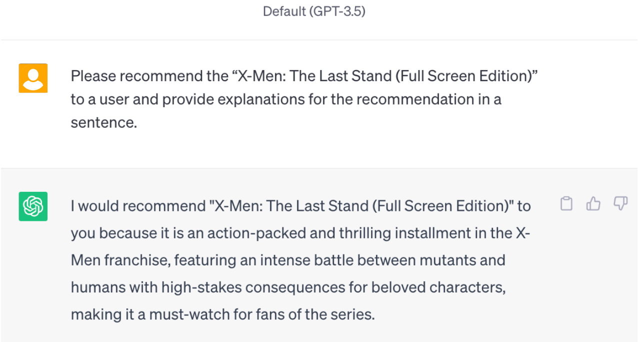
Abstract

Large language model-based explainable recommendation (LLM-based ER) systems can provide remarkable human-like explanations and have widely received attention from researchers. However, the original LLM-based ER systems face three low-quality problems in their generated explanations, i.e., lack of personalization, inconsistency, and questionable explanation data. To address these problems, we propose a novel LLM-based ER model denoted as `LLM2ER` to serve as a backbone and devise two innovative explainable quality reward models for fine-tuning such a backbone in a reinforcement learning paradigm, ultimately yielding a fine-tuned model denoted as `LLM2ER-EQR`, which can provide high-quality explanations. `LLM2ER-EQR` can generate personalized, informative, and consistent high-quality explanations learned from questionable-quality explanation datasets. Extensive experiments conducted on three real-world datasets demonstrate that our model can generate fluent, diverse, informative, and highly personalized explanations.

Introduction

Explainable recommendation (ER) systems aim to provide high-quality explanations to help users understand the recommendations and make decisions. According to the phenomenon reported in an exhaustive survey of explanation quality (Lu et al. 2023), generating human-like explanations can significantly improve the adoption rate of recommended items. Among various explanation forms, such as tags (Yan et al. 2020), reasoning paths (Wang et al. 2019) and images (Chen et al. 2019a), the textual explanation generated by large language models (LLMs) has attracted increasing attention due to their remarkable human-like natural language generation capabilities.

Intuitively, a straightforward solution for an LLM-based ER system is to feed an ER-related instructional prompt to an LLM. In Figure 1, we show an ER process for a movie named “X-Men” using a simple instruction with ChatGPT. We can observe that the explanation given by ChatGPT assumes that the user is already a fan of the X-Men series and contains a lot of terms that appear in the movie, which may be difficult to understand for the user who, in fact, is not familiar with the movie. Based on this example, we can conclude that such an explanation suffers from *low-quality* problem, making this explanation unable to improve user satisfaction. Moreover, the widely recognized issue of hallucination (Ji et al. 2023) in LLM can lead to factual inaccuracies within explanations, thereby compounding the low-quality problem. Therefore, directly using an LLM to fulfill the ER task cannot be one-size-fits-all.



Default (GPT-3.5)

Please recommend the “X-Men: The Last Stand (Full Screen Edition)” to a user and provide explanations for the recommendation in a sentence.

I would recommend "X-Men: The Last Stand (Full Screen Edition)" to you because it is an action-packed and thrilling installment in the X-Men franchise, featuring an intense battle between mutants and humans with high-stakes consequences for beloved characters, making it a must-watch for fans of the series.

Figure 1: The explanation generated by ChatGPT (GPT-3.5) with an ER-related instructional prompt on July 10, 2023.

With the above insightful study of LLM-based ER, we analyze that the causes of the low-quality problem are three-fold.

Cause 1: *Prompts without integrating personalized information trigger a lack of personalization in explanations.* LLM is usually pre-trained on generic data and lacks personalized information from users, and thus it may generate explanations that do not match user preferences when personalized user–item information is not integrated into the prompt, i.e., the model input. Existing studies (Geng et al. 2022; Li, Zhang, and Chen 2023) construct personalized prompts with personalized information to guide LLM in generating explanations. However, the personalization degree in such generated explanations is limited, because they incorporate user and item information separately rather than

[*] Corresponding author

incorporating collaborative user-item information.

Cause 2: *Generating information-overloaded explanations reduces consistency.* LLMs generate informative but lengthy explanations, which contain a considerable amount of irrelevant information about the item features that a user does not care about, resulting in a worse user experience. Existing studies (Chen et al. 2019b; Hada and Shevade 2021) leverage both user's and item's historical reviews to learn the user preferences and item features to guide LLM in generating precise explanations. However, the user preferences implied in the user's historical reviews partially match the item features implied in the corresponding item's historical reviews. Jointly leveraging the historical reviews from the dual sides will mix in some words that do not match user preferences and item features simultaneously, resulting in a reduction in the precision of the generated explanations.

Cause 3: *Lack of sufficient high-quality explanation data for fine-tuning limits the adaptability of LLM to the ER task.* To better adapt to the ER task, LLM requires fine-tuning with explanation data. Existing studies (Hada and Shevade 2021; Li, Zhang, and Chen 2023) fine-tune LLM by aligning paired generated explanation and review from the same user-item pair, i.e., denoting the review as the ground truth explanation. However, the review dataset is questionable, and the ground truth corresponding to a generated explanation is not necessarily of high quality. For example, we often encounter hollow reviews that lack substantial information, and more importantly, based on such reviews, it is difficult to fine-tune the model to generate convincing explanations.

In order to improve the quality of explanations for LLM-based ER, we tend to apply reinforcement learning from human feedback (RLHF) as a training paradigm to obtain an ER-oriented LLM, which is a popular technique to alleviate the hallucinations and low-quality problems of LLMs. However, RLHF requires handcrafted evaluation with expensive manual efforts, which is impractical in recommender systems. Fortunately, the recommendation system contains a wealth of implicit information that can evaluate the quality of explanations from multiple perspectives, allowing us to fine-tune the LLM in an unsupervised manner.

Based on the above analysis, in this paper, we propose a novel LLM-based ER backbone named `LLM2ER` and fine-tune such a backbone in a reinforcement learning paradigm with two novel explainable quality reward models, where the fine-tuned model is named `LLM2ER-EQR`. `LLM2ER` model a concept graph extracted from reviews to achieve the following two purposes: (1) predict the rating based on the heterogeneous graph model and map the rating to the user's sentiment for the target item; (2) infer the reasoning paths between target user-item pair from the concept graph to collect personalized and consistent candidate concepts to improve the precision of explanations (for addressing **Causes 1** and **2**). The `LLM2ER-EQR` additionally includes two reward models to further enhance the explanation quality: (1) concept consistent reward model leverages sentiment-wise candidate concepts to preserve paired user preferences and item features in the generated explanations based on contrastive learning (for addressing **Cause 2**); (2) high-quality alignment reward model aligns the generated explanations to unpaired high-quality explanations based on the generative adversarial network (for addressing **Cause 3**).

To the best of our knowledge, this is the first work to fine-tune an LLM in a reinforcement learning paradigm for explainable recommendations. Our main contributions are summarized as follows: (1) *Effective model design*: we propose a novel fine-tuned LLM-based ER model `LL2ER-EQR` to address three low-quality problems in ER systems; (2) *Novel fine-tuning strategy*: we devise an efficient and feasible RL-based fine-tuning strategy for unsupervised fine-tuning of LLMs that can generate high-quality explanations without a handcrafted evaluation; (3) *Extensive experiments*: we conduct extensive experiments on three real-world datasets, which demonstrate our model significantly outperforms the state-of-the-art methods and can generate fluent, diverse, informative, and highly personalized explanations.

Related Work

With natural language processing technology having achieved remarkable performance in text generation, ER based on text generation has received increasing attention because of its good human-like language generation capabilities. The early studies propose methods to generate explanations based on pre-defined templates (Wang et al. 2018) or association rules (Gao et al. 2019), which require extensive manual labor and cannot assemble diverse, personalized, convincing explanations. Subsequently, there are methods (Chen et al. 2019b; Li, Zhang, and Chen 2020; Hu et al. 2022; Zhang et al. 2023) to generate synthetic text explanations by recurrent neural network-based (RNN-based) language models. However, their models' training processes are confronted with a shortage of sufficient training samples, i.e., explanation texts, resulting in a lack of robustness in generated explanations. In addition, the RNN-based language models are not trained on a vast corpus, which makes the fluency of generated explanations questionable.

Recently, a burgeoning body of research started to study LLM-based ER models. (Li, Zhang, and Chen 2021; Geng et al. 2022; Li, Zhang, and Chen 2023) mainly focus on designing prompts to guide LLMs to directly generate explanations. However, LLMs face the hallucination problem resulting in generating low-quality explanations. Some studies address part of low-quality problems, such as questionable review data and personalization, by controlled text generation (Hada and Shevade 2021), personalized variational autoencoder (Cai and Cai 2022; Wang et al. 2023), and retrieval model (Xie et al. 2023). So far, there is no research that comprehensively addresses the low-quality problems of explanations generated by LLM-based ER models.

In addition, there are a series of studies orthogonal to our work, i.e., review-based recommendation models (Zheng, Noroozi, and Yu 2017; Shuai et al. 2022), which integrates the explanation process into the recommendation model to improve recommendation performance rather than generating explainable text and is not our main focus.

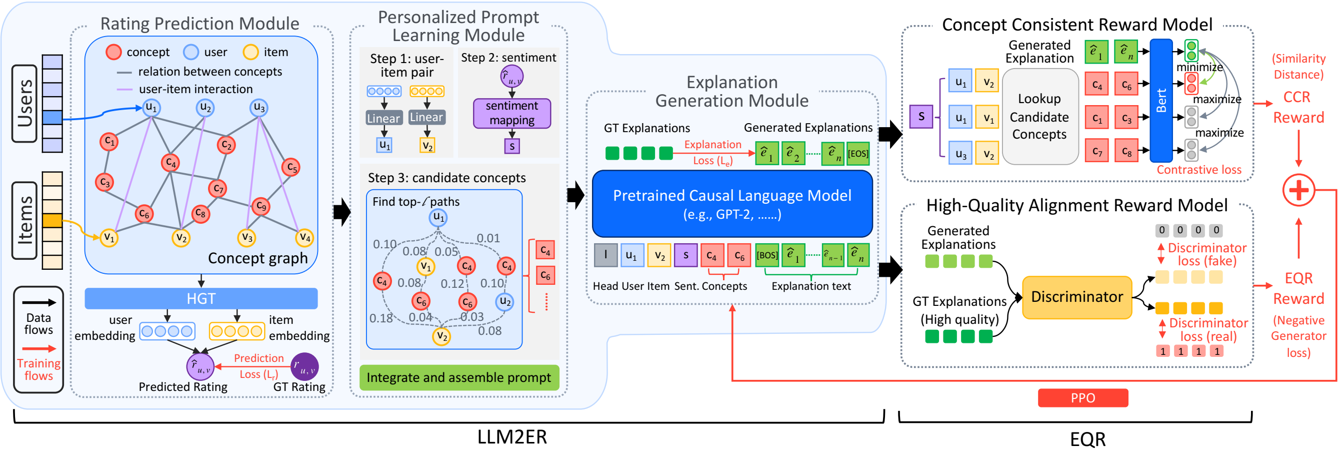


Figure 2: The architecture of LLM2ER-EQR. The left part is the backbone for the ER task named LLM2ER, and the right part is the explainable quality reward models named EQR.

Preliminaries and Task Formulation

Preliminaries

The decision process for the RL problem can be defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$, where $\mathcal{S}$ denotes a finite state space, $\mathcal{A}$ denotes a finite set of actions, $\mathcal{T}(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ is a state transition function that defines the next state $\mathbf{s}'$ given the current state $\mathbf{s}$ and action $\mathbf{a}$, and $\mathcal{R}(\mathbf{s}, \mathbf{a})$ is a reward function. A policy $\pi(\mathbf{a}|\mathbf{s})$ determines an action $\mathbf{a}$ given the current state $\mathbf{s}$.

Reward learning enables the application of reinforcement learning (RL) to fine-tune an LLM. InstructGPT (Ouyang et al. 2022) apply an RL training paradigm using Proximal Policy Gradient (PPO) (Schulman et al. 2017) as the fine-tuning strategy. Specifically, in InstructGPT, the RL agent π_ϕ^{RL} is initialized with the supervised pre-trained LLM π^{SFT}, which aims to maximize the following combined objective function f_{RL}:

$$f_{\mathrm{RL}} = E_{(x,\hat{e})\sim D_{\pi_\phi^{\mathrm{RL}}}} [\underbrace{\mathcal{R}_\theta(x,\hat{e})}_{\text{reward}} - \beta \underbrace{\log(\pi_\phi^{\mathrm{RL}}(\hat{e}\mid x)/\pi^{\mathrm{SFT}}(\hat{e}\mid x))]}_{\text{KL penalty}} - \underbrace{\gamma\, E_{x\sim D_{\text{pretrain}}}[-\log(\pi_\phi^{\mathrm{RL}}(x))]}_{\text{pre-training loss}}, \tag{1}$$

where x is the prompt, $\hat{e}$ is the generated text, the reward $\mathcal{R}_\theta(x, \hat{e})$ is calculated by a reward model, D_{pretrain} is the pre-training distribution, and the coefficients β and γ control the KL penalty and pre-training loss, respectively.

Explainable Recommendation Task

We formulate the ER task as follows.

Input. The input consists of the user set $\mathcal{U}$, the item set $\mathcal{V}$ and the concept graph:

(1) each **user** is represented by its ID $u \in \mathcal{U}$ and each **item** is represented by the item ID $v \in \mathcal{V}$;

(2) the **rating** $r_{u,v} \in \mathbb{R}_{>0}$ is a positive value given by user u to item v. Let $\Omega = \{(u, v) :$ user u rates item $v\}$. Note that unavailable ratings are represented by 0, namely, $r_{u,v} = 0$ where $(u, v) \notin \Omega$. Specifically, we treat Ω^{train}, Ω^{test} as the training and test dataset, respectively.

(3) the **explanation** $e_{u,v}$ is a short text, which is extracted from a review of the user-item pair (u, v). For each explanation denoted as $e_{u,v}$, we assemble a corresponding set of concepts represented as $\{c\}_{u,v}$. These concepts are systematically extracted from the explanation text. Note that the elements within this concept set are not arbitrary; they are keywords pertinent to recommendation explanations, including organizations, careers, characteristics, companies, brands, services, products, etc. These keywords are instrumental in mirroring user preferences and item attributes. Specifically, we extract several concepts $\{c\}_{u,v}$ according to the following two steps: (i) extract 1-gram to 4-grams from the explanation; (ii) match them to Microsoft Concept Graph and get the matched words as concepts.

(4) the **concept graph** $\mathcal{G}$ is a heterogeneous graph, which includes user nodes $\mathcal{U}$, item nodes $\mathcal{V}$, and concept nodes $\mathcal{C}$. In order to enrich concept relations, we add the 1-hop neighbors of all concepts into $\mathcal{C}$. We organize the above information in the form of knowledge graph $\mathcal{G} = \{(h, r, t)|h, t \in \mathcal{C} \cup \mathcal{U} \cup \mathcal{V}, r \in \mathcal{M} \cup \mathcal{I}\}$, where $\mathcal{M}$ is the relation set from Microsoft Concept Graph, $\mathcal{I}$ is the interaction set of user-explanation-item interactions. Note that the concept graph is only constructed based on the training dataset.

Output. Given a user-item pair $(u, v) \notin \Omega_{\text{train}}$, our model predicts: (1) the **rating** $\hat{r}_{u,v}$ and (2) the **explanation** $\hat{e}_{u,v}$ for the target user-item pair (u, v) .

Note that at the training stage, the input data comprise users, items, ground truth ratings, ground truth explanations, and the concept graph, while during the testing stage, only users, items, and the concept graph are exposed to the model.

Method

The architecture of LLM2ER-EQR is presented in Figure 2. In the following subsections, we first present an LLM-based explainable recommendation model named LLM2ER as the backbone for the ER task. After that, to further improve the quality of explanations, we devise the fine-tuning process of the LLM-based backbone based on a reinforcement learning paradigm, where the backbone denotes the action model. We elaborate on two types of reward models corresponding to address consistency problem and questionable explanation data problem. Finally, we present the whole training process of `LLM2ER-EQR` .

Pre-trained Language Models, Prompt Learning, and Explanation Generation

We introduce three modules of the LLM2ER backbone according to the process of an ER task.

Collaborative Concept-based Rating Prediction Module. For each user-item pair, we process this module in the following two steps. Firstly, we learn the user and item embedding based on the concept graph, which contains rich user preferences and item features. Specifically, we adopt HGT (Hu et al. 2020) to aggregate the interaction and concept information from heterogeneous concept graphs and obtain the corresponding node embeddings $\mathbf{h}$. Secondly, we employ a multi-layer perception (MLP) to predict the rating that the user u rates the item v as follows:

$$\hat{r}_{u,v} = \text{MLP}([\mathbf{h}_u, \mathbf{h}_v]), \tag{2}$$

where $\mathbf{h}_u$ and $\mathbf{h}_v$ are the user and item node embedding, $[\cdot, \cdot]$ denotes the concatenation of vectors.

For the rating prediction task, we aim to minimize the mean squared error loss between ground truth ratings and the predicted ones as follows:

$$\mathcal{L}_r = \frac{1}{|\Omega^{\text{train}}|} \sum_{u,v \in \Omega^{\text{train}}} (r_{u,v} - \hat{r}_{u,v})^2. \tag{3}$$

Personalized Prompt Learning Module. Plugging personalized information into the LLM by prompt learning can control the LLM to generate personalized explanations. To construct such a personalized prompt, we create an ER prompt template following the instruction-based prompt schema (Geng et al. 2022), which contains detailed ER task descriptions and personalized information as follows:

Prompt head I provides detailed instructions in the natural language format for the ER task.

User-item embeddings pair $(\mathbf{p}_u, \mathbf{q}_v)$ are important identifiers for personalization. We use linear projection to conform the dimension of user and item embeddings to that of the LLM token embedding, as follows:

$$(\mathbf{p}_u, \mathbf{q}_v) = (\text{Linear}(\mathbf{h}_u), \text{Linear}(\mathbf{h}_v)). \tag{4}$$

Sentiment $s_{u,v}$ reflects the user's attitude about the item, which is indicated by rating. Following (Li, Zhang, and Chen 2020), we map the rating to “negative” if the rating is less than 3 (5-scale rating), and vice versa.

Candidate concepts $\{\hat{c}\}_{u,v}$ are a set composed of distinct concepts collected from reasoning paths for consistency. Benefiting from the attention mechanism in HGT, we infer the reasoning paths from the target user u to target item v by performing a weighted search on the concept graph. Specifically, we adopt beam search to heuristically search the top-l paths with the highest cumulative attention weights and with no more than h hops. To limit prompt length, we stop the search once the size of the candidate concept set $|\{\hat{c}\}_{u,v}|$ reaches 10.

For each target user u and target item v, we assemble a personalized prompt $x_{u,v} = [I, \mathbf{p}_u, \mathbf{q}_v, s_{u,v}, \{\hat{c}\}_{u,v}]$, where all the text are tokenized by the LLM's tokenizer.

Explanation Generation Module. We adopt a pre-trained causal language model, such as GPT-2, to generate the explanation $\hat{e}_{u,v} = (\hat{e}_{u,v,1}, \hat{e}_{u,v,2}, \ldots, \hat{e}_{u,v,n})$, where n is the explanation length. For the i-th decoding, we have

$$\hat{e}_{u,v,i} = \arg\max_{\hat{e}_{u,v,i}} p_{\mathcal{M}}(\hat{e}_{u,v,i} | x_{u,v}, \hat{e}_{u,v,<i}), \tag{5}$$

where decoding output $\hat{e}_{u,v,i}$ is conditioned on both input prompt $x_{u,v}$ and previous outputs $\hat{e}_{u,v,<i}$.

For the explanation generation task, we aim to the negative log-likelihood loss between the ground truth explanation and the generated explanation as follows:

$$\mathcal{L}_e = \frac{1}{|\Omega^{\text{train}}|} \sum_{u,v \in \Omega^{\text{train}}} \frac{1}{n} \sum_{t=1}^{n} -\log p(e_{u,v,i}), \tag{6}$$

where $p(e_{u,v,i})$ is the predicted probability of token $e_{u,v,i}$.

Concept Consistent Reward Model (CCR)

In order to further alleviate low-quality problems of generated explanations, we fine-tune the backbone `LLM2ER` in an RL paradigm followed by InstructGPT (Ouyang et al. 2022), with an explainable quality reward model.

To improve the generated explanation consistent with user preferences and item features, we propose a concept consistent reward model (CCR) based on BERT (Kenton and Toutanova 2019). The CCR calculates the similarity distance between the pair of generated explanations and the corresponding sentiment-wise candidate concepts as the reward. Specifically, CCR first encodes the generated explanations $\hat{e}_{u,v}$ and candidate concepts $\{\hat{c}\}_{u,v}$ by BERT, and then calculates the cosine similarity of the averaged word embedding over the words of $\hat{e}_{u,v}$ and $\{\hat{c}\}_{u,v}$. The reward is calculated by:

$$\mathcal{R}_{\text{CCR}}(\hat{e}_{u,v}, \{\hat{c}\}_{u,v}) = \text{cosin}(\text{avg}(\text{BERT}(\hat{e}_{u,v})), \text{avg}(\text{BERT}(\{\hat{c}\}_{u,v}))). \tag{7}$$

Before we plug the CCR model into the RL paradigm, we need to train it by reward learning. We construct contrastive learning to train the CCR model from the perspective of consistency. Specifically, to form the positive pair, we take the generated explanation $\hat{e}_{u,v}$ of a target user-item pair (u, v) and the corresponding candidate concepts $\{\hat{c}\}_{u,v}$ as a positive pair. To form the negative pair, we select some user-item pairs as negative samples $\mathcal{N}_{u',v'}$ and pair $\hat{e}_{u',v'}$ with the candidate concepts $\{\hat{c}\}_{u',v'}$ of each negative user-item pair (u', v'). The negative user-item pairs are selected by the following rules: (1) the user or item in such user-item pair has one that matches the target user-item pair; (2) the negative user-item pairs have the same sentiment as that in the target user-item pair. Finally, following (Gao, Yao, and Chen 2021), we train the CCR model by minimizing the contrastive loss as:

$$\mathcal{L}_c = \frac{1}{|\Omega^{\text{train}}|} \sum_{(u,v) \in \Omega^{\text{train}}} \log \frac{\exp(r_{\text{CCR}}(\hat{e}_{u,v}, \{\hat{c}\}_{u,v}))}{\sum_{(u',v') \in \mathcal{N}_{u,v}} \exp(r_{\text{CCR}}(\hat{e}_{u',v'}, \{\hat{c}\}_{u',v'}))}. \tag{8}$$

High-Quality Alignment Reward Model (HQAR)

Because the ground truth corresponding to the generated explanation is not necessarily of high quality, we propose a high-quality alignment reward model (HQAR) based on generative adversarial network (GAN) (Goodfellow et al. 2014) to align the explanations generated by LLM2ER to unpaired high-quality explanations that contain rich information during the training process.

Specifically, we reformat LLM2ER and HQAR into a GAN framework, where LLM2ER is the generator and HQAR is the discriminator. The primary objective of LLM2ER within this GAN framework is to generate high-quality explanations with the intention of deceiving the HQAR. Conversely, the HQAR's goal is to discern these high-quality explanations with accuracy. This dynamic establishes a competitive interaction between the LLM2ER and HQAR, central to the effectiveness of the LLM2ER. Note that HQAR includes an LLM with the same structure as the LLM in LLM2ER. We feed the generated explanation $\hat{e}_{u,v}$ to the HQAR, and the HQAR outputs the probability p that the generated explanation came from a high-quality explanations dataset. We denote the negative generative loss as the high-quality alignment reward, as follows:

$$\mathcal{R}_{\text{HQAR}}(\hat{e}_{u,v}) = -l_g(\hat{e}_{u,v}) = -\sum_{i=1}^{n} log(1 - p_{\hat{e}_{u,v,i}}), \qquad (9)$$

where i is the token index.

We train the HQAR model by following two steps. Firstly, we collect a high-quality explanation dataset $\mathcal{H}$ as a real sample dataset. For each ground truth explanation in the training dataset, we define a concept proportion rate of each ground truth explanation to measure the proportion of words in the explanation belonging to the corresponding candidate concepts. The concept proportion rate is calculated as $o_e = |\{c\}_{u,v} \cap e_{u,v}|/n$, where n is the explanation length. Then we set a threshold δ and select those explanations with $o_e \geq \delta$ to form $\mathcal{H}$. Note that we set the δ empirically. Secondly, we feed the generated explanation $\hat{e}_{u,v}$ and high-quality explanation h sampled from $\mathcal{H}$ to HQAR (discriminator), and HQAR outputs the probability p that the explanation came from $\mathcal{H}$. We adopt a token-level binary cross-entropy loss as a discriminator loss to train the HQAR, which can be calculated as:

$$\mathcal{L}_d = -\frac{1}{|\Omega^{\text{train}}| + |\mathcal{H}|}(\sum_{(u,v)\in\Omega^{\text{train}}} \sum_{i=1}^{n} \log(1 - p_{\hat{e}_{u,v,i}}) + \sum_{h\in\mathcal{H}} \sum_{i=1}^{n} \log p_{h_i}), \qquad (10)$$

where i is the token index.

Model Training

The training process of the LLM2ER-EQR consists of four steps. The first step is training the collaborative concept-based rating prediction module by minimizing the $\mathcal{L}_r$. The second step is training the LLM2ER by minimizing the $\mathcal{L}_{\text{LLM2ER}}$, where $\mathcal{L}_{\text{LLM2ER}} = \lambda_r \mathcal{L}_r + \lambda_e \mathcal{L}_e$. The third step is training the two reward models, i.e., CCR and HQAR, under a frozen LLM2ER by minimizing the $\mathcal{L}_{\text{EQR}}$, where $\mathcal{L}_{\text{EQR}} = \lambda_c \mathcal{L}_c + \lambda_d \mathcal{L}_d$. The fourth step is fine-tuning the LLM2ER. Specifically, training the LLM2ER under two frozen reward models by maximizing the combined objective in Eq. (1), where the reward function is defined as follows:

$$\mathcal{R}(x_{u,v}, \hat{e}_{u,v}) = \lambda_{\text{CCR}} \mathcal{R}_{\text{CCR}}(\hat{e}_{u,v}, \{\hat{c}\}_{u,v}) + \lambda_{\text{HQAR}} \mathcal{R}_{\text{HQAR}}(\hat{e}_{u,v}), \qquad (11)$$

where $\{\hat{c}\}_{u,v}$ is in prompt $x_{u,v}$. Note that the λ_r, λ_e, λ_c, λ_d, λ_{CCR} and λ_{HQAR} are coefficients for balancing the importance of different loss terms. We perform the third and fourth steps alternately until the fine-tuned LLM2ER and the two reward models reach Nash equilibrium (Goodfellow et al. 2014).

Experiment

In this section, we present extensive experiments to answer the following question:

Q1: How does LLM2ER-EQR perform on explanation generation task?

Q2: How do LLM2ER-EQR's key components contribute to its performance?

Q3: How do the concept selection affect the performance of LLM2ER-EQR?

Q4: How about the quality of explanations generated by LLM2ER-EQR?

Experimental Settings

Dataset Descriptions and Collection. To evaluate the effectiveness of LLM2ER-EQR, we adopt three benchmark recommendation datasets, which are publicly available explainable contents and vary in terms of domain, size, and sparsity. The three datasets are from Amazon (Movie & TV)[1], and Yelp (2019)[2], and TripAdvisor[3], respectively, and their corresponding recommendation explanation data are collected from the GitHub repository[4] of (Li, Zhang, and Chen 2021). To ensure the quality of the concept graphs, we then filter out the rare concepts and domain-dependent frequent concepts. Statistics of the datasets are shown in Table 1.

Following the previous study (Li, Zhang, and Chen 2020, 2021; Wang et al. 2023), each dataset is randomly divided into training, validation, and testing sets with a ratio of 8:1:1. We repeat all experiments 5 times independently, with each iteration involving a re-division of the dataset. The mean of test performance is reported.

Comparison Methods. To demonstrate the effectiveness of LLM2ER-EQR, we compare seven LM-ER baselines, all of which generate explanations by the language model. **CAML** (Chen et al. 2019b) and **NETE** (Li, Zhang, and Chen 2020) are two conventional GRU-based ER models to generate explanations and template-controlled explanations, respectively. **ReXPlug** (Hada and Shevade 2021) is a state-of-the-art ER model and applies a plug-and-play language model to generate controlled text explanations. **PETER** (Li, Zhang, and Chen 2021) is a state-of-the-art LLM-based ER model to enhance explanation generation by prompt learning. **PEVAE** (Cai and Cai 2022) is a state-of-the-art LLM-based ER model extending a hierarchical variational autoencoder to overcome the data sparsity. **CVAEs** (Wang et al.

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

[2] https://www.yelp.com/dataset

[3] https://www.tripadvisor.com

[4] https://github.com/lileipisces/PETER

Dataset	Amazon (Movies & TV)	Yelp (2019)	Trip-Advisor
# Users	7,506	27,147	9,765
# Items	7,316	18,172	5,429
# exp.	432,075	1,189,056	296,118
# con.	10,141	14,657	10,948
# Triplets	7,223,673	31,417,433	6,992,065
Avg. # exp./user	57.56	43.80	30.32
Avg. # exp./item	59.06	65.43	54.54
Avg. # con./exp.	7.33	11.46	9.73

Table 1: Statistics of the datasets. The "exp." is the abbreviation of "explanations" and the "con." is the abbreviation of "concepts"

2023) is a state-of-the-art LLM-based ER model to extract adequate characteristics for controllable generation. **PRAG** (Xie et al. 2023) is a state-of-the-art LLM-based ER model for addressing factual hallucinations problem.

Evaluation Metrics. We evaluate the performance of explanation generation from the following two perspectives.

In terms of the text quality perspective, we adopt two widely used evaluation metrics, i.e., BLEU (Chen and Cherry 2014) and ROUGE (Lin 2004). Specifically, we report the results of BLEU-3 and BLEU-4, denoting the smoothed BLEU scores for 3-grams and 4-grams, respectively. We present the result of ROUGE-1 and ROUGE-L, denoting the ROUGE scores for 1-grams and longest common subsequence, respectively.

In terms of the explanation quality perspective, we adopt three metrics for comprehensive evaluation. From a personalization aspect, we adopt Distinct-1 and Distinct-2 for sentence-wise diversity and propose a new metric named Concept Overlapping Ratio (COR) for word-level diversity. COR measures the overlap of concepts in any two generated explanations with the same item. COR is calculated as:

$$\begin{aligned} \mathrm{COR}(v) &= \frac{1}{|\mathcal{U}_v^{\mathrm{test}}|(|\mathcal{U}_v^{\mathrm{test}}|-1)} \sum_{u\in\mathcal{U}_i^{\mathrm{test}},u'\in\mathcal{U}_i^{\mathrm{test}}} \frac{|(\hat{e}_{u,v}\cap\{c\}_{u,v})\cap(\hat{e}_{u',v}\cap\{c\}_{u',v})|}{|(\hat{e}_{u,v}\cap\{c\}_{u,v})\cup(\hat{e}_{u',v}\cap\{c\}_{u',v})|}, \\ \mathrm{COR} &= \frac{1}{|\mathcal{V}^{\mathrm{test}}|}\sum_{v\in\mathcal{V}^{\mathrm{test}}}\mathrm{COR}(v), \end{aligned} \tag{12}$$

where $\mathcal{U}_v^{\mathrm{test}}$ denotes the set of users that have interactions with v in the testing data, and $\mathcal{V}^{\mathrm{test}}$ denotes the item set in testing data. A lower COR indicates a smaller overlap between explanations and, thus, a higher diversity. From a consistency aspect, we propose Concept Matching Ratio (CMR) to measure how many concepts are contained in the generated explanation, which is calculated as follows:

$$\mathrm{CMR} = \frac{1}{|\Omega^{\mathrm{test}}|}\sum_{(u,v)\in\Omega^{\mathrm{test}}}\sum_{c\in\{c\}_{u,v}}\delta(c\in\hat{e}_{u,v}), \tag{13}$$

where $\delta(\cdot)$ is an indicator function that is true when concept c matches 1-gram to 4-grams in $\hat{e}_{u,v}$. Higher CMR indicates better performance.

Personalized Prompt. The personalized ER prompt used in our LLM2ER is as follows, which contains four placeholders, i.e., sentiment, candidate concepts, user embedding, and item embedding.

ER prompt template:

Prompt head: Please generate recommendation explanation.
Query input: Based on the {sentiment: positive\negative } sentiment and candidate concepts including {candidate concepts, e.g., user preferences, item features}, please provide explanations to recommend {item embedding} to {user embedding}.
Query output: {explanation}

For each user-item pair, we assemble the sentiment, candidate concepts, user embedding, and item embedding into the above ER prompt template to construct the entire personalized prompt.

Quantitative on Explanation Tasks (for Q1)

We compare LLM2ER-EQR with seven LM-ER methods. The evaluation results of the generated explanations on three datasets are shown in Table 2, which shows LLM2ER-EQR consistently outperforms baselines on all metrics. From the results, we can demonstrate the effectiveness of LLM2ER-EQR from the following three aspects. Firstly, LLM2ER-EQR generates explanations with the highest text quality, as it achieves the highest BLEU and ROUGE scores, with an average improvement of 5.496% and 4.835%, respectively, compared to the best-performing baseline. Secondly, LLM2ER-EQR exhibits the ability to generate diverse and personalized explanations, as is reflected by the highest Distinct scores and COR scores among all baselines, with an average improvement of 3.940% and 12.756% compared to the best-performing baseline, respectively. Thirdly, in terms of consistency, we observe that LLM2ER-EQR achieves the highest CMR scores with an average improvement of 5.843% over the best-performing baseline, which demonstrates the CCR model in LLM2ER-EQR has the capability to alleviate the consistency problem in the generated explanations.

Ablation Studies (for Q2)

To investigate the effectiveness of LLM2ER-EQR's key components, we experiment on three simplified variants of our own model, i.e., LLM2ER, LLM2ER-CCR and LLM2ER-HQAR, only backbone, only with concept consistent reward model, and only with high-quality alignment reward model, respectively. As shown in Table 2, the absence of any key component will lead to a decline in performance. Specifically, LLM2ER-EQR outperforms LLM2ER on all eight explainable metrics, which empirically demonstrates the importance of improving explanation quality. LLM2ER-EQR outperform LLM2ER-HQAR with average improvements of 9.460% on CMR, which demonstrates LLM2ER-EQR has the strong ability to preserve the consistent concepts that reflect both user pref-

Categories	Methods	BLEU (%)		ROUGE (%)		Distinct (%)		NEW Defined	
		BLEU-1	BLEU-4	ROUGE-1	ROUGE-L	Distinct-1	Distinct-2	COR	CMR
Amazon (Movie & TV)									
LM-ER	CAML	15.834	1.439	15.686	12.104	18.256	61.254	0.078	2.078
	NETE	13.847	1.186	14.891	11.880	14.854	48.431	0.099	0.759
	RexPlug	15.940	1.436	17.406	13.040	17.167	57.233	0.101	2.040
	PETER	15.850	1.403	16.514	13.395	18.135	60.290	0.082	2.042
	PEVAE	16.288	1.476	17.051	13.345	18.650	62.710	0.077	1.894
	PRAG	15.756	1.389	16.417	13.316	19.003	64.420	0.069	2.146
	CVAEs	17.319	1.504	17.114	14.536	18.714	62.367	0.080	2.023
Our Model	LLM2ER	16.378	1.193	16.874	14.176	18.400	64.982	0.066	2.171
	LLM2ER-HQAR	16.973	1.237	17.669	14.614	19.089	66.228	0.071	2.196
	LLM2ER-CCR	17.319	1.564	18.029	14.747	18.622	65.631	**0.057**	2.301
	LLM2ER-EQR	**17.571***	**1.572****	**18.291****	**15.157****	**19.370****	**67.044****	0.058**	**2.353****
Improvement[1]		1.457%	4.520%	5.089%	4.276%	1.931%	4.073%	16.403%	9.647%
Yelp (2019)									
LM-ER	CAML	13.533	1.283	15.080	11.837	16.677	56.830	0.090	2.422
	NETE	15.414	1.322	18.040	14.226	13.345	45.466	0.127	1.756
	RexPlug	15.978	1.368	19.960	15.668	16.760	59.680	0.093	2.860
	PETER	14.989	1.026	18.720	14.246	16.274	52.852	0.080	2.805
	PEVAE	16.219	1.524	18.867	15.012	17.210	60.570	0.081	2.542
	PRAG	16.612	1.146	20.747	15.789	16.901	60.045	0.071	2.897
	CVAEs	17.804	1.511	20.064	16.839	16.764	59.979	0.090	2.549
Our Model	LLM2ER	18.075	1.523	21.427	16.827	16.879	59.916	0.071	2.800
	LLM2ER-HQAR	17.841	1.307	21.487	16.730	17.617	63.050	0.076	2.746
	LLM2ER-CCR	17.968	1.653	21.640	16.849	16.897	62.317	**0.063**	2.953
	LLM2ER-EQR	**18.013***	**1.662****	**21.693****	**16.901***	**17.707****	**64.150****	0.064**	**2.995****
Improvement		1.170%	9.042%	4.563%	0.365%	2.888%	5.911%	11.276%	3.365%
TripAdvisor									
LM-ER	CAML	16.385	1.151	17.468	14.832	19.388	63.583	0.060	3.510
	NETE	15.783	1.138	17.403	14.521	15.616	52.130	0.075	1.694
	RexPlug	16.726	1.216	20.472	15.939	20.051	67.263	0.064	3.978
	PETER	17.023	1.197	20.445	16.084	21.122	65.843	0.059	3.670
	PEVAE	16.806	1.307	18.911	15.408	20.380	68.444	0.055	3.571
	PRAG	17.179	1.213	20.634	16.232	20.956	69.226	0.049	4.008
	CVAEs	18.748	1.331	20.902	16.829	21.926	68.842	0.059	3.586
Our Model	LLM2ER	18.589	1.272	20.952	16.689	21.758	68.325	0.067	3.735
	LLM2ER-HQAR	18.642	1.433	21.113	16.774	22.495	71.398	0.055	3.734
	LLM2ER-CCR	19.983	1.382	21.532	17.981	21.782	69.993	**0.044**	4.178
	LLM2ER-EQR	**20.040****	**1.436****	**22.196****	**18.032****	**22.794****	**72.601****	0.045**	**4.189****
Improvement		6.894%	9.892%	7.572%	7.146%	3.959%	4.875%	10.587%	4.516%

[1] Improvement of LLM2ER-EQR over the best-performing baselines (underlined numbers indicate best-performing baselines' results).
** and * respectively indicate the statistical significance for $p < 0.01$ and $p < 0.05$ via Student's t-test.

Table 2: Performance comparison of all methods of generated explanations on three datasets.

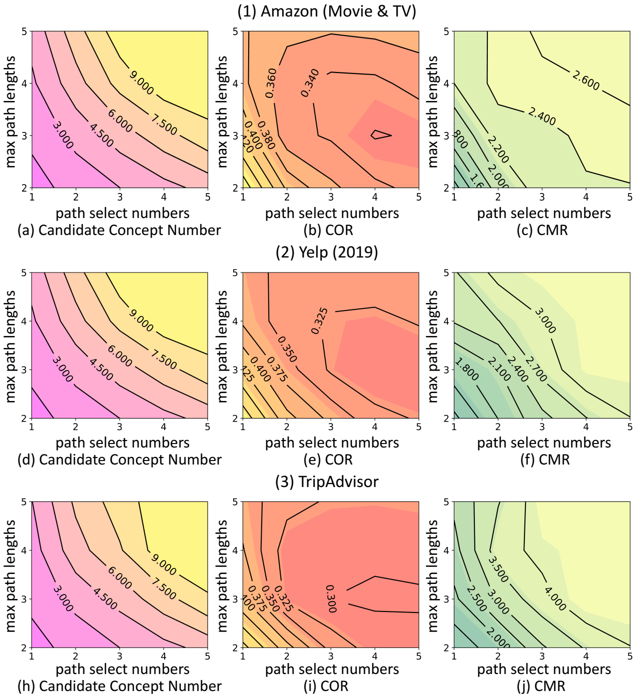
erences and item features in the generated explanations. LLM2ER-EQR outperform LLM2ER-CCR with average improvements of 3.713% on Distinct score, which indicates the LLM2ER-EQR model's capability to infuse informative terms into unpaired insubstantial explanations through adept alignment, thereby enriching the diversity of explanations.

Effectiveness of Concept Selection (for Q3)

To study the effectiveness of involving concepts, we evaluate the performance of LLM2ER-EQR with different max path lengths and path select numbers. Figure 3 shows the results on the three datasets. From Figure 3 (a), (d), and (h), we can observe that both the max path length and the path select number affect the number of concepts. Jointly considering both Figure 3 (b) and (c) for the Amazon (Movie & TV) dataset, the performance reaches its peaks with a max path length of 3 and a path select number of 5. A similar trend is observed in the Yelp (2019) and TripAdvisor datasets. From this result, we can conclude that an excessive quantity of candidate concepts leads to an increase in COR, thereby diminishing the degree of personalization, while an inadequate number of candidate concepts results in diminished CMR. Therefore, we select the max path length as 3 and the path select number as 5 for our model.

Qualitative Case Study (for Q4)

We present some concrete cases to show the improvement of generated explanations of our model. For comparison, we show three cases of explanations generated by baselines and LLM2ER-EQR in the Amazon dataset (c.f., Figure 4). For each user-item pair in the three cases, we display concepts that the user and item most frequently mentioned in



(a) Candidate Concept Number (b) COR (c) CMR (d) Candidate Concept Number (e) COR (f) CMR (h) Candidate Concept Number (i) COR (j) CMR

Figure 3: Performance on different max path lengths and path select numbers on three datasets.

the training dataset as user preferences and item features, respectively. Overall, compared with all baselines, our explanations are more informative, similar to ground-truth explanations, and exhibit fluency and coherence. Specifically, by jointly comparing the generated explanations between case 1 and case 2, when recommending the same item to different users, LLM2ER−EQR generates personalized and diverse explanations, aligning effectively with users' preferences. Moreover, our explanations contain the largest number of user preferences and item features, demonstrating that LLM2ER−EQR can effectively maintain the consistency of generated explanations.

Discussion and Conclusion

Legacies of Language Models. It is worth noting that a large part of our framework relies on the pre-trained LLMs (Brown et al. 2020). This means that LLM2ER naturally inherits the advantages and disadvantages of the pre-trained LLM. Specifically, the advantage is that LLM2ER-EQR can utilize the implicit knowledge existing in the pre-trained LLM to generate explanations, while the disadvantage is that it may introduce risks, such as potentially producing offensive language, propagating social biases and stereotypes, and leaking private information (Weidinger et al. 2021). To alleviate such a problem, we have made efforts in both training and inference stages: (1) during the training stage, the LLM is fine-tuned by the cleaned ER datasets free of harmful information to guide it to generate high-quality explanations; (2) during the inference stage, we construct prompts with reasoned concepts about user preferences and item attributes to guide LLM generate explanations in the direction without harmful information. Nonetheless, we must admit that when actually deploying our model in a real scenario, it is necessary to take post-processing to further prevent the display of harmful information to users.

1	**Item**: X-Men **Item feature**: screen, character, action, wolverine, scenes, intense, future **User preferences**: **screen**, **character**, **action**, **enjoy**, **wolverine**, **score**, **tear**
Ground-truth	There will be a spin-off movie featuring **wolverine**. He has a good **character** and intense **actions**.
CAML	Intense movie. **Wolverine's character** is good. **Enjoy** it.
NETE	The movie is very fun and intense, and the main **character** is very good.
ReXPlug	I like a **wolverine** movie because he is the most popular **character** on **screen**.
PETER	**Wolverine** is cool and **characters** are good. A good movie for **action** fans.
PEVAE	The **wolverine** movie is **enjoyable**, his **actions** and character are good.
PRAG	The movie excelled with intense **character**, iconic **wolverine**, groundbreaking **action**.
CVAEs	The movie is a **character**-driven **action** movie with intense **wolverine** scenes.
LLM2ER	The movie crafted a good **character** and presented intense **action** scenes.
LLM2ER-EQR	The movie crafted an impressive **wolverine** **character** and presented intense **action** scenes.
2	**Item**: X-Men **Item feature**: screen, character, action, wolverine, scenes, intense, future **User preferences** : **director**, **definite**, **action**
Ground-truth	One can **definitely** tell that this **director** has a long future in **directing** **action** movies.
CAML	It is an **action** movie **directed** by a good **director**.
NETE	The movie is a very good **action** movie and the story is very intense.
ReXPlug	The good **director** and wolverine 's character makes the movie a must-see.
PETER	This intense movie is **directed** by a skilled **action** **director**.
PEVAE	The **director** made a good **action** movie.
PRAG	The movie has definite **action**, deep character and futuristic scenes.
CVAEs	This movie feature intense **action**, good characters, and a skillful **director**.
LLM2ER	The excellent **director** achieved a satisfying blend of scenes and character.
LLM2ER-EQR	The **action** **director** has a vision that blends futuristic scenes and captivating character.
3	**Item**: Star Wars **Item feature**: George Lucas, theme, previous, war **User preferences** : **popular**, **wonderful**, **brilliant**, **previous**
Ground-truth	It 's **brilliant** that George Lucas kept some of the themes from the **previous** films going here.
CAML	I like the war theme of the **popular** and **wonderful** movie.
NETE	The movie is very **popular** and the theme is about war.
ReXPlug	George Lucas genius shines in this movie and the theme links to **previous** ones.
PETER	This war movie is worth watching and it is a continuation of the **previous** theme.
PEVAE	It is a **popular** movie, maintaining themes from **previous** films.
PRAG	This movie features **brilliant** theme, widespread **popularity** and **wonderful** story telling.
CVAEs	This movie kept **previous** theme, being highly **popular**.
LLM2ER	George Lucas astutely kept the theme of the **previous** films to create this war film.
LLM2ER-EQR	George Lucas kept the theme of the **previous** films, creating a **wonderful** and **popular** film.

Figure 4: Explanations generated by baselines and LLM2ER−EQR. The item features and user preferences mentioned in the generated explanation are highlighted.

Conclusions and Future Work. In this paper, we propose a novel LLM-based ER backbone LLM2ER and fine-tune such a backbone as LLM2ER−EQR in an RL paradigm with two explainable quality reward models, to address three low-quality problems, i.e., personality, consistency, and poor review quality in explanations. Extensive experiments demonstrate the superiority of LLM2ER over the state-of-the-art methods for the explainable recommendation tasks. In the future, we plan to fully exploit LLM2ER 's potential by migrating to multimodal recommended forms.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2022YFF0902704), Leading Expert of "Ten Thousands Talent Program" of Zhejiang Province (No.2021R52001), and the cooperation project of MYbank, Ant Group.

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language Models are Few-shot Learners. In *Proc. of NeurIPS*, 1877–1901.

Cai, Z.; and Cai, Z. 2022. PEVAE: A Hierarchical VAE for Personalized Explainable Recommendation. In *Proc. of SIGIR*, 692–702.

Chen, B.; and Cherry, C. 2014. A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU. In *Proc. of WMT*, 362–367.

Chen, X.; Chen, H.; Xu, H.; Zhang, Y.; Cao, Y.; Qin, Z.; and Zha, H. 2019a. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proc. of SIGIR*, 765–774.

Chen, Z.; Wang, X.; Xie, X.; Wu, T.; Bu, G.; Wang, Y.; and Chen, E. 2019b. Co-attentive multi-task learning for explainable recommendation. In *Proc. of IJCAI*, 2137–2143.

Gao, J.; Wang, X.; Wang, Y.; and Xie, X. 2019. Explainable recommendation through attentive multi-view learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3622–3629.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proc. of EMNLP*, 6894–6910.

Geng, S.; Liu, S.; Fu, Z.; Ge, Y.; and Zhang, Y. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proc. of RecSys*, 299–315.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Proc. of NeurIPS*, 1–9.

Hada, D. V.; and Shevade, S. K. 2021. Rexplug: Explainable recommendation using plug-and-play language model. In *Proc. of SIGIR*, 81–91.

Hu, Y.; Liu, Y.; Miao, C.; Lin, G.; and Miao, Y. 2022. Aspect-guided syntax graph learning for explainable recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous graph transformer. In *Proc. of WWW*, 2704–2710.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12): 1–38.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 4171–4186.

Li, L.; Zhang, Y.; and Chen, L. 2020. Generate neural template explanations for recommendation. In *Proc. of CIKM*, 755–764.

Li, L.; Zhang, Y.; and Chen, L. 2021. Personalized Transformer for Explainable Recommendation. In *Proc. of ACL*, 1–11.

Li, L.; Zhang, Y.; and Chen, L. 2023. Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, 41(4): 1–26.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Association for Computational Linguistics.

Lu, H.; Ma, W.; Wang, Y.; Zhang, M.; Wang, X.; Liu, Y.; Chua, T.-S.; and Ma, S. 2023. User Perception of Recommendation Explanation: Are Your Explanations What Users Need? *ACM Transactions on Information Systems*, 41(2): 1–31.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. In *Proc. of NeurIPS*, 27730–27744.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 1–12.

Shuai, J.; Zhang, K.; Wu, L.; Sun, P.; Hong, R.; Wang, M.; and Li, Y. 2022. A review-aware graph contrastive learning framework for recommendation. In *Proc. of SIGIR*, 1283–1293.

Wang, L.; Cai, Z.; de Melo, G.; Cao, Z.; and He, L. 2023. Disentangled CVAEs with Contrastive Learning for Explainable Recommendation. In *Proc. of AAAI*, 13691–13699.

Wang, N.; Wang, H.; Jia, Y.; and Yin, Y. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *Proc. of SIGIR*, 165–174.

Wang, X.; He, X.; Cao, Y.; Liu, M.; and Chua, T.-S. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proc. of SIGKDD*, 950–958.

Weidinger, L.; Mellor, J.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.-S.; Cheng, M.; Glaese, M.; et al. 2021. Ethical and Social Risks of Harm from Language Models. *arXiv preprint arXiv:2112.04359*, 1–64.

Xie, Z.; Singh, S.; McAuley, J.; and Majumder, B. P. 2023. Factual and informative review generation for explainable recommendation. In *Proc. of AAAI*, 13816–13824.

Yan, S.; Chen, X.; Huo, R.; Zhang, X.; and Lin, L. 2020. Learning to build user-tag profile in recommendation system. In *Proc. of CIKM*, 2877–2884.

Zhang, J.; Chen, X.; Tang, J.; Shao, W.; Dai, Q.; Dong, Z.; and Zhang, R. 2023. Recommendation with Causality enhanced Natural Language Explanations. In *Proceedings of the ACM Web Conference 2023*, 876–886.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proc. of WSDM*, 425–434.