

Coherency Improved Explainable Recommendation via Large Language Model

Shijie Liu^{1*}, Ruixing Ding^{1*}, Weihai Lu^{2*}, Jun Wang¹, Mo Yu³, Xiaoming Shi¹, Wei Zhang^{1†}

¹East China Normal University, ²Peking University, ³WeChat AI, Tencent
karrich128@gmail.com, zhangwei.thu2011@gmail.com

Abstract

Explainable recommender systems are designed to elucidate the explanation behind each recommendation, enabling users to comprehend the underlying logic. Previous works perform rating prediction and explanation generation in a multi-task manner. However, these works suffer from incoherence between predicted ratings and explanations. To address the issue, we propose a novel framework that employs a large language model (LLM) to generate a rating, transforms it into a rating vector, and finally generates an explanation based on the rating vector and user-item information. Moreover, we propose utilizing publicly available LLMs and pre-trained sentiment analysis models to automatically evaluate the coherence without human annotations. Extensive experimental results on three datasets of explainable recommendation show that the proposed framework is effective, outperforming state-of-the-art baselines with improvements of 7.3% in explainability and 4.4% in text quality.

Code — <https://github.com/karrich/CIER>

Introduction

Recommendation systems provide personalized suggestions to maximize user engagement and satisfaction based on historical interactions and preferences (Zhang et al. 2019), showing significant potential and technological value. Recently, to relieve the concerns regarding trustworthiness due to the inherent lack of transparency and explainability, explainable recommendation systems have been introduced (Zhang and Chen 2020; Zhang et al. 2022). These systems elucidate the rationale behind each recommendation, enabling users to comprehend the underlying logic. This enhanced understanding empowers users to make informed decisions and fosters greater trust in the system’s suggestions.

Current works on explainable recommendation systems generate ratings and provide corresponding explanations (Ni et al. 2017; Sun et al. 2020; Li, Zhang, and Chen 2021; Cheng et al. 2023). Specifically, the rating prediction and explanation generation modules are jointly learned in a multi-task learning manner, sharing a common hidden representa-

*These authors contributed equally.

†Corresponding author.

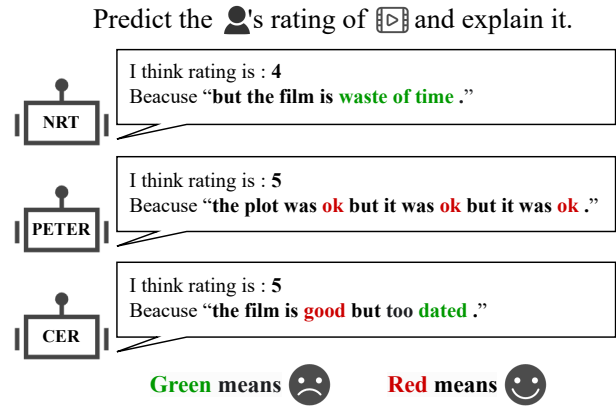


Figure 1: Explanations generated by NRT, PETER, and CER for an example from Amazon Movies.

tion layer but having individual output layers. Despite improvements in explanations, these methods suffer from incoherence between predicted ratings and explanations. As shown in Figure 1, NRT (Li et al. 2017) and PETER (Li, Zhang, and Chen 2021) generate inconsistent explanations. This inconsistency arises because these two tasks only share hidden layer representation, and explanation generation does not explicitly include rating information. To enhance the coherency, CER (Raczyński, Lango, and Stefanowski 2023) proposes explanation-based rating estimation, obtaining explanation embeddings through max pooling of generated text embeddings and minimizing the distance between the explanation and the corresponding rating vectors. Despite the improved coherency, CER suffers from two issues: (1) CER utilizes a small-sized transformer as the backbone, which limits the generative performance. (2) CER struggles to enforce coherence due to poor sentence embedding, as it relies on max pooling of pre-trained word embeddings, which fails to capture rich contextual information (Nee-lakantan et al. 2022; Wang et al. 2023). As such, CER fails to generate coherent explanations, as reflected in the figure.

Recently, the revolutionary progress in large language models (LLM) (Zeng et al. 2022; OpenAI 2023; Touvron et al. 2023) has catalyzed substantial technological trans-

formations in natural language generation and reshaped its foundation. Inspired by LLMs, we propose using them as the backbone model to predict ratings and generate explanations for recommendation systems. LLMs produce fluent and accurate ratings and explanations, addressing the first issue. To tackle the second issue, we propose generating ratings and explanations in a pipeline manner, similar to next-token prediction, which is suitable for decoder-based LLMs.

Specifically, an LLM is fine-tuned with LoRA (Hu et al. 2022) to predict ratings, which are subsequently transformed into rating vectors, while explanations are generated using both user and item information in conjunction with rating vectors. The generation process utilizes the rating as input for the LLM, enhancing the coherency through its in-context learning capability. Meanwhile, training techniques such as rating smoothing, curriculum learning, and multi-task learning are employed to enhance performance, with experiments demonstrating their effectiveness.

Besides, coherency evaluation is crucial yet challenging. Current methods can be divided into manual and automatic evaluations. Manual evaluation, while effective, is labor-intensive and impractical at scale. To address this, a study (Raczyński, Lango, and Stefanowski 2023) proposes using a binary classifier trained on manually annotated data for automatic evaluation. Despite its high efficiency, this automated metric relies heavily on the quality and quantity of the annotated data, which is time-consuming and costly. To overcome these limitations, we propose utilizing GPT-4 (Achiam et al. 2023) and a pre-trained sentiment analysis model (NLP Town 2023) to assess coherency without additional manual annotations. GPT-4 excels in advanced natural language understanding, while the BERT-based pre-trained model is tailored for sentiment classification in product reviews, making both well-suited for our purposes.

The main contributions are as follows:

- To generate more coherent explanations, we propose a framework, named CIER (Coherency-Improved Explainable Recommendation), which initially predicts a rating with LLMs and subsequently leverages the rating to generate an explanation.
- For a more streamlined assessment of coherency between ratings and explanations, we propose to employ LLMs and pre-trained sentiment analysis models.
- We conduct extensive experiments to demonstrate the effectiveness of the proposed framework against strong baselines, and experimental results show that training techniques can further improve the results.

Related work

Explainable Recommender Systems

In recent years, more and more research has focused on how to provide good explanations for recommendations to enhance system effectiveness and user satisfaction. Various explanation styles include topical word clouds (Al-Taie and Kadry 2014), highlighted images (Chen et al. 2019), knowledge graphs (Fu et al. 2020), and automatically generated textual explanations (Li, Zhang, and Chen 2021). The latter is of particular interest, as textual explanations are more

easily comprehended by users, particularly non-expert users, and more informative than pre-defined templates.

In this work, we focus on generating high-quality explanatory texts while providing accurate recommendations. Our proposed CIER framework aims to address the flaw of inconsistencies between recommendations and natural language explanations provided by existing methods (Li, Zhang, and Chen 2021; Li et al. 2017; Li, Zhang, and Chen 2023; Raczyński, Lango, and Stefanowski 2023; Yang et al. 2021; Zhang et al. 2023; Sun et al. 2020).

LLMs for Explainable Recommendation

With the advancement of natural language generation techniques, several studies have employed Recurrent Neural Networks (e.g., Long Short-Term Memory (Hochreiter and Schmidhuber 1997), Gated Recurrent Unit (Cho et al. 2014)), unpretrained Transformer (Vaswani et al. 2017) and pre-trained language models (e.g., BERT (Devlin et al. 2019)) for generating explanations. Pre-trained large language models are initially introduced in PEPLER (Li, Zhang, and Chen 2023) to enhance the performance of explanation generation. Although PEPLER utilizes prompt-based transfer learning with GPT-2 (Radford et al. 2019), it fails to structure training data in a manner suitable for instruction tuning, thereby limiting the system’s ability to produce high-quality explanations.

Our proposed CIER framework is designed to harness the language capabilities of LLMs to advance the field of explainable recommender systems.

Explainable Recommendation Evaluation Metrics

Previous works mostly rely on perplexity and overlapping-based metrics such as Distinct-N (Li et al. 2016), Rouge score (Lin 2004), and BLEU score (Papineni et al. 2002), to evaluate against the ground truth explanations. However, none of these metrics assess how truthfully the generated explanations reflect the rating predictions.

The studies (Raczyński, Lango, and Stefanowski 2023; Yang et al. 2021) introduce some automatic methods for evaluating the consistency between predictions and explanations. However, the reliance on the manual rules and quality of annotations significantly impacts the effectiveness and reliability of the evaluation process, which also raises concerns about reproducibility. To address these limitations, we introduce a new automatic evaluation method that uses publicly available pre-trained language models to assess rating-explanation coherence.

Methodology

The overview of the proposed method CIER is depicted in Figure 2, with three modules, rating prediction, soft rating to word embedding (SR2WE), and explanation generation. In what follows, we first provide the problem formulation, then introduce the details and training techniques of CIER, and finally describe the proposed automatic evaluation method for assessing the coherence.

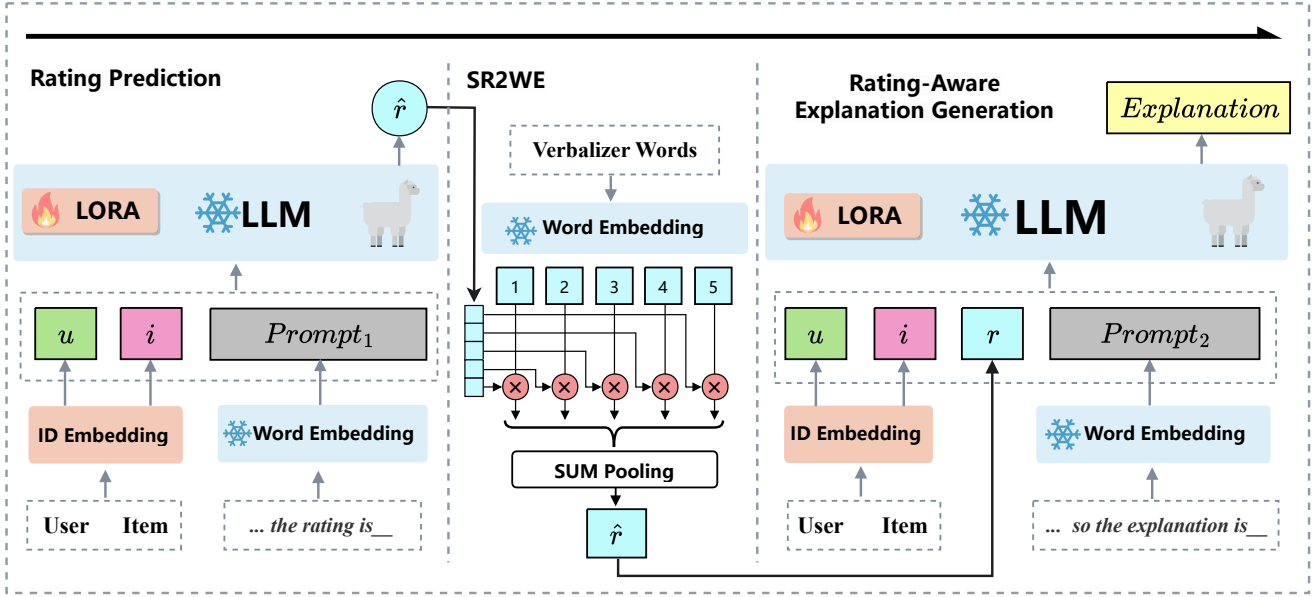


Figure 2: The overview framework of CIER. (a) Rating Prediction: aiming to predict users’ ratings of items based on LLM. (b) SR2WE: embedding the predicted soft rating into the LLM word embedding space. (c) Rating-Aware Explanation Generation: using the predicted ratings as context to generate explanations related to the ratings.

Problem Formulation

Given a pair of user u and item i , the objective is to jointly predict a rating $r_{u,i}$ and generate an explanation $E_{u,i}$ that justifies this rating. The rating $r_{u,i}$ is a score from 1 to 5 that reflects the user u ’s preference towards the item i . The explanation $E_{u,i}$ is a sequence of tokens from a predefined vocabulary \mathcal{V} that provides a personalized verbalizer.

Proposed Method CIER

Rating Prediction The objective of the rating prediction task is to forecast the rating that a specified user u would assign to a candidate item i , which is akin to the goal of most recommendation tasks. To construct a unified framework for both the rating prediction and explanation generation tasks, we employ LLaMA2-7B as the backbone for CIER and use a corresponding verbalizer (Hu et al. 2021) specifically for the rating prediction component.

The verbalizer V^r is a fixed mapping from numeric ratings to their word representations, defined as $V^r = \{1: \text{“1”}, 2: \text{“2”}, 3: \text{“3”}, 4: \text{“4”}, 5: \text{“5”}\}$. This design facilitates consistency in the rating prediction process. The probability assigned by the model to each word in V corresponds to the probability of each respective rating:

$$\hat{r}_{u,i} = f([u, i, p_1, \dots, p_m]), \quad (1)$$

where p represents the prompt, f is the LLM, m is the prompt length, and $\hat{r}_{u,i}$ is the predicted probability of each rating. Then the rating is obtained by weighted summation:

$$\hat{r}_{score} = \sum_{x=1}^{|r|} \hat{r}_{u,i,x} \cdot x, \quad (2)$$

where $|r|$ is the number of rating classes, $\hat{r}_{u,i,x}$ is the probability of rating x , and $\sum_{x=1}^{|r|} \hat{r}_{u,i,x} = 1$.

Soft Rating to Word Embedding For a given rating, the hard rating embedding directly uses the corresponding word embedding in the verbalizer. However, hard ratings have less information than soft ratings, so we try to embed soft-rating into the word embedding space, which is defined as follows:

$$\mathbf{s}_{r_{u,i}} = \sum_{x=1}^{|r|} \hat{r}_{u,i,x} \cdot \text{Embedding}_{LLM}(V^r(x)), \quad (3)$$

where Embedding_{LLM} is the word embedding layer of the LLM, and $V^r(x)$ is the corresponding word of rating x in the verbalizer. At this point, we have obtained the semantic representation of the predicted rating, which encapsulates the uncertainty and distribution features of user u ’s preference towards item i .

Rating-Aware Explanation Generation The rating-aware explanation generation module aims to generate an explanation based on given u , i , and $r_{u,i}$.

The process is formulated as follows:

$$E_{u,i} = f([u, i, \mathbf{s}_{r_{u,i}}, p_1, \dots, p_j]), \quad (4)$$

where p represents the prompt, f is the LLM, j is the prompt length, $\mathbf{s}_{r_{u,i}}$ is the rating embedding from SR2WE module, and $E_{u,i}$ is the generated explanation.

Training Techniques

To balance efficiency and performance, we conduct Lora tuning for LLM. In addition, three training techniques are utilized in this work for better performance, i.e., rating smoothing, curriculum learning, and multi-task learning.

Rating Smoothing Using a probability distribution over possible ratings to obtain the rating embedding in the inference phase offers several potential benefits. However, training the model exclusively on ground-truth ratings introduces a notable disparity between the training phase and inference.

To address this, we introduce a rating smoothing technique that is inspired by label smoothing but incorporates enhancements tailored to our specific scenario. Traditional label smoothing distributes probability across all categories, potentially diluting the model’s sensitivity to user-specific ratings. In rating prediction, adjacent ratings contain similar sentiments, so our proposed rating smoothing prevents over-smoothing by limiting the impact to ratings that are numerically adjacent to the ground truth ratings (called neighboring ratings). Specifically, with a probability γ , the original one-hot distribution of rating $r_{u,i}$ is transformed to:

$$r_{u,i,x}^{\text{modified}} = \begin{cases} 1 - \alpha & \text{if } x = r_{u,i} \\ \frac{\alpha}{k} & \text{if } x \in \mathcal{N}_{r_{u,i}}^k \\ 0 & \text{others,} \end{cases} \quad (5)$$

where $\alpha \in [0, \frac{k}{k+1}]$, $\mathcal{N}_{r_{u,i}}^k$ denotes the set of k neighboring ratings of $r_{u,i}$. Regarding to the selection of the smoothing technique, various possibilities are explored and the proposed rating smoothing is intuitive and experimentally proven to be effective.

Training with Curriculum Learning Previous methods struggle to capture explanatory keywords that reflect users’ interests in explanations, showing low explainability. To address this problem, we introduce a keyword generation task to help the model identify item features (e.g. lobby, location) that the user cares about in explanations.

Inspired by curriculum learning, we propose a training strategy that allows models to build foundational knowledge before tackling more intricate problems. Specifically, we devise a linear transition mechanism that dynamically adjusts the data allocation between the keyword generation and explanation generation tasks during training. The transition probability $P(t)$ represents the likelihood of the data point used for explanation generation task in batch t :

$$P(t) = \frac{t}{T}, \quad (6)$$

where T denotes the total number of training batches. During each batch, data points are probabilistically assigned to either task based on a random number n generated from a uniform distribution over $[0, 1]$. The assignment is determined by comparing n with $P(t)$:

$$\text{Task}(t) = \begin{cases} \text{Task}_{\text{explanation}} & \text{if } n < P(t) \\ \text{Task}_{\text{keyword}} & \text{if } n \geq P(t). \end{cases} \quad (7a) \quad (7b)$$

The training process initially focuses on predicting the keywords of explanations, gradually shifting towards generating complete explanations. This approach retains foundational knowledge while integrating the complexities of explanation generation.

Figure 3 shows the specific instructions and prompts used. During the keyword training process, “explanation” in the

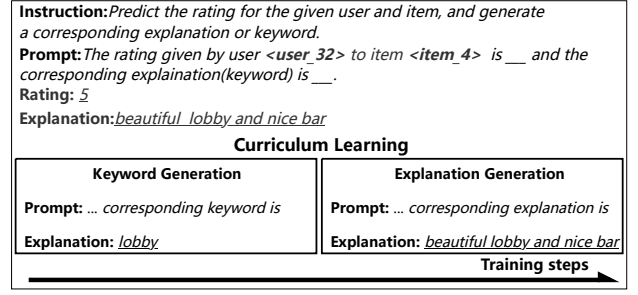


Figure 3: Instructions and prompts for curriculum learning.

prompt will be replaced with “keyword”, and the target will be replaced from a complete explanation to the key words in the explanation.

Multi-task Learning The cross-entropy loss (CE) is utilized as the loss function for rating prediction:

$$\mathcal{L}_r = -\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} \sum_{x=1}^{|r|} r_{u,i,x} \log(\hat{r}_{u,i,x}), \quad (8)$$

where \mathcal{T} denotes the training set and $r_{u,i,x}$ is the probability of the ground-truth rating being x . We use the Negative Log-Likelihood (NLL) as the loss function for the text (i.e., explanation or keyword) generation, computing the mean over user-item pairs in the training set.

$$\mathcal{L}_e = \frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} \frac{1}{|E_{u,i}|} \sum_{t=1}^{|E_{u,i}|} -\log c_{|S|-|E_{u,i}|+t}^{e_t}. \quad (9)$$

The probability $c_i^{e_t}$ is offset by $|S| - |E_{u,i}| + t$ positions because the generated text is placed at the end of the sequence.

We integrate rating prediction and text generation into a multi-task learning framework. The objective function is defined as follows:

$$\mathcal{J} = \min_{\Theta = \{\Theta_{Lora}, \Theta_U, \Theta_I\}} (\mathcal{L}_e + \lambda \mathcal{L}_r), \quad (10)$$

where Θ denotes all the trainable parameters in the model, including the parameters of Lora modules, i.e., Θ_{Lora} , and the parameters of ID Embeddings, i.e., Θ_U and Θ_I . The hyperparameter λ is used to balance the learning between the explanation generation task and the rating prediction task. It is worth noting that these two tasks are performed in a pipeline manner like next-token prediction, thus suitable for decoder-based LLMs.

Automatic Coherence Evaluation

To address heavy reliance on high-quality annotated data in the previous approach (Raczyński, Lango, and Stefanowski 2023), we employ publicly available pre-trained language models, specifically GPT-4 and bert-base-multilingual-uncased-sentiment (NLP Town 2023), to automatically assess the rating-explanation coherency. GPT-4 has recently demonstrated remarkable performance across various tasks, leading to its widespread use as an evaluator (Sun et al.

Datasets	Yelp	Amazon	TripAdvisor
#users	27,147	7,506	9,765
#items	20,266	7,360	6,280
#records	1,293,247	441,783	320,023
#features	7,340	5,399	5,069

Table 1: Statistics of the datasets.

2024; Zhou et al. 2023). Meanwhile, the BERT-based model is specifically designed for sentiment analysis in product reviews, making it particularly suitable for our purposes.

For GPT-4, a prompt is utilized to provide clear guidelines on how sentiment should match each rating level and an instruction is used to make it respond with “Yes” or “No” based on the coherency between ratings and explanations. The percentage of coherent rating-explanation pairs identified by GPT-4 serves as a performance metric. Specifically, the “gpt-4o” model is utilized to evaluate randomly sampled 500 predictions from each model.

Bert-base-multilingual-uncased-sentiment is applied to predict the sentiment rating of explanations for all predictions. Given the influence of personalized factors on rating predictions and the individual biases across different datasets, coherency is defined as the predicted sentiment rating deviating by no more than one point from the given rating, defined as follows:

$$\text{Coherency} = \begin{cases} 1 & \text{if } |y - \hat{y}| \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (11a)$$

$$(11b)$$

where y represents the rating provided by explainable recommendation model, and \hat{y} represents that from the sentiment classification model.

Experiments

Experimental Setting

Dataset To validate the effectiveness of our method, we conducted experiments on three publicly available datasets and their splits (Li, Zhang, and Chen 2020). Each dataset is randomly divided into training, validation, and test sets in an 8:1:1 ratio five times. The three datasets are from TripAdvisor (hotel), Amazon (movies & TV), and Yelp (restaurant). Each record in the dataset consists of a user ID, an item ID, a rating on a scale of 1 to 5, an explanation, and item features. The explanations are sentences extracted from user reviews. Features are attributes of items extracted from the explanation, e.g., *lobby*, which represent aspects users care about, and we consider them as the keyword of explanations. The dataset statistics are shown in Table 1. The available datasets and keyword extraction tools are provided by Sentires (Zhang et al. 2014).

Evaluation Metrics To evaluate the performance of rating prediction, we utilize two commonly used metrics: Root Mean Square Error (**RMSE**) and Mean Absolute Error (**MAE**) to measure the deviation between predicted ratings and ground truth ratings.

For explanation performance, we measure the generated explanations from two main perspectives: text quality and

explainability. For the text quality, we use **BLEU** (Papineni et al. 2002) and **ROUGE** (Lin 2004), which are common metrics in natural language generation tasks. Specifically, we use BLEU-1 and BLEU-4 metrics to evaluate the precision, the recall-scores of ROUGE-1 and ROUGE-2 to evaluate the recall, and the f1-score of ROUGE-L for comprehensive evaluation. For the text explainability, we use additional indicators proposed by (Li, Zhang, and Chen 2020) to measure explainability: Feature Matching Ratio (**FMR**), Feature Coverage Ratio (**FCR**), Feature Diversity (**DIV**), and Unique Sentence Ratio (**USR**).

To measure the coherence between explanations and predicted ratings, we perform manual and automated evaluations. For manual evaluation, we follow CER (Raczyński, Lango, and Stefanowski 2023) to annotate the coherence with two independent human annotators. For automatic annotation, we use our proposed automatic evaluation method.

Baselines To evaluate the explainability performance, we compare the following explanation methods:

NRT (Li et al. 2017) utilizes GRU (Cho et al. 2014) to jointly predict ratings and generate explanations using user and item IDs as input.

Att2Seq (Dong et al. 2017) is an explanation generation model based on LSTM (Hochreiter and Schmidhuber 1997).

PETER (Li, Zhang, and Chen 2021) is a powerful multi-layer Transformer (Vaswani et al. 2017) model that simultaneously predicts ratings and generates explanations.

CER (Raczyński, Lango, and Stefanowski 2023) proposes a module that estimates the discrepancy between predicted ratings and explanation-based ratings to enhance rating-explanation coherency.

PEPLER (Li, Zhang, and Chen 2023) leverages the advanced capabilities of GPT-2 through prompt-based transfer learning and regularization loss.

ERRA (Cheng et al. 2023) is a multi-layer Transformer with aspect enhancement and retrieval enhancement. Since the code is incomplete, we directly use its results in its paper.

For the evaluation of recommendation performance, in addition to NRT, PETER, and CER, we also use three traditional models as baselines:

SVD++ (Koren 2008) integrates implicit feedback from users to enhance the latent factors.

DeepCoNN (Zheng, Noroozi, and Yu 2017) learns item properties and user behavior from review text.

NARRE (Chen et al. 2018) applies the attention mechanism to the rating prediction task.

For evaluating coherence, we use PETER, CER, and CIER-M as baselines. CIER-M means that CIER masks the context (predicted ratings) when generating explanations.

Implementation Details

All the experiments are conducted on an NVIDIA H800 GPU. We utilize the validation set to tune hyper-parameters for each dataset, and subsequently present the average evaluation metrics computed across 5 data splits on the testing set. We load LLaMA2-7B from HuggingFace as the backbone of our proposed model, utilizing BPE (Sennrich, Haddow, and Birch 2016) for vocabulary construction. To ensure

	Explainability				Text Quality				
	FMR↑	FCR↑	DIV↓	USR↑	B-1↑	B-4↑	R-1↑	R-2↑	R-L↑
Yelp									
NRT (Li et al. 2017)	6.65	11.96	1.77	16.02	11.36	0.65	12.35	1.29	10.39
Att2Seq (Dong et al. 2017)	7.08	14.24	1.72	17.65	11.47	0.69	12.46	1.35	10.40
PETER (Li, Zhang, and Chen 2021)	8.09	13.80	<u>1.65</u>	8.47	9.68	0.62	11.63	1.26	10.24
CER (Raczyński, Lango, and Stefanowski 2023)	8.05	15.00	1.59	9.67	10.03	0.65	11.72	1.29	10.27
PEPLER (Li, Zhang, and Chen 2023)	8.11	21.04	1.73	20.32	10.94	0.67	12.05	1.36	10.41
ERRA* (Cheng et al. 2023)	\	\	\	\	10.71	0.73	\	1.36	10.82
CIER (Curriculum Learning)	8.71	53.84	1.67	32.63	11.78	0.83	13.02	1.59	10.90
Two-Stage Learning	8.61	<u>52.46</u>	1.67	31.30	<u>11.62</u>	<u>0.82</u>	<u>12.89</u>	<u>1.57</u>	<u>10.85</u>
Vanilla Learning	<u>8.62</u>	52.19	1.70	<u>32.48</u>	11.39	0.79	12.78	1.54	<u>10.85</u>
Amazon									
NRT (Li et al. 2017)	11.13	5.67	2.38	14.57	12.62	0.89	13.82	1.88	11.24
Att2Seq (Dong et al. 2017)	11.11	8.22	2.17	22.12	12.86	0.92	13.88	1.87	11.19
PETER (Li, Zhang, and Chen 2021)	11.60	9.12	2.20	13.30	12.38	1.00	13.45	1.94	11.29
CER (Raczyński, Lango, and Stefanowski 2023)	11.47	10.25	2.09	14.72	12.02	1.02	13.23	1.92	11.05
PEPLER (Li, Zhang, and Chen 2023)	11.88	34.07	2.26	24.87	12.57	1.03	13.83	1.92	11.31
CIER (Curriculum Learning)	12.45	51.80	<u>2.08</u>	46.99	13.55	1.15	14.61	2.09	11.70
Two-Stage Learning	<u>12.21</u>	<u>51.11</u>	2.01	<u>50.26</u>	13.43	<u>1.18</u>	14.50	<u>2.12</u>	<u>11.67</u>
Vanilla Learning	12.00	50.84	<u>2.08</u>	50.92	<u>13.53</u>	1.23	<u>14.58</u>	2.13	11.65
TripAdvisor									
NRT (Li et al. 2017)	5.76	14.15	3.09	18.29	14.85	0.96	15.07	1.98	12.24
Att2Seq (Dong et al. 2017)	5.78	10.61	<u>2.92</u>	10.33	15.16	0.97	15.17	1.97	12.22
PETER (Li, Zhang, and Chen 2023)	6.47	13.72	<u>3.03</u>	9.60	15.97	1.04	15.94	2.25	12.64
CER (Raczyński, Lango, and Stefanowski 2023)	6.97	12.99	3.14	9.18	15.59	1.09	15.89	2.19	12.75
PEPLER (Li, Zhang, and Chen 2023)	7.36	19.91	3.35	24.29	15.06	1.02	14.92	2.03	12.21
ERRA* (Cheng et al. 2023)	\	\	\	\	16.13	1.06	\	2.15	13.17
CIER (Curriculum Learning)	8.08	<u>36.99</u>	3.05	<u>29.86</u>	17.00	1.31	17.07	2.54	13.40
Two-Stage Learning	<u>7.89</u>	39.08	3.00	31.80	<u>16.54</u>	<u>1.28</u>	<u>16.70</u>	<u>2.45</u>	<u>13.33</u>
Vanilla Learning	7.73	36.60	2.86	27.63	16.45	1.25	16.66	2.40	13.31

Table 2: Results of explanation. B-1, B-4, R-1, R-2 and R-L represent the scores of BLUE-1, BLEU-4, ROUGE-1, ROUGE-2 and ROUGE-L, respectively. BLEU, ROUGE, FMR, FCR, and USR are presented as percentage (%), while the others are absolute values. The best values in the table are represented in bold, and the second-best values are represented with underlines. Stars* indicate that the results of this method are from its paper.

Method	Explanation	rating
Truth	swimming pool was small and shallow	1
NRT	the bed was comfortable and the room was comfortable	3
PETER	the hotel is a little dated but the rooms are very small	3
CER	the resort is a bit dated but the hotel is a bit dated	1
CIER	the pool is a bit small and the gym is a bit small	1

Table 3: Example generated by CIER and baselines.

fair comparisons, we apply BPE to all baseline models and set the max explanation length to 20 BPE tokens. For CIER, λ is set to 0.1 and γ to 0.2, selected through grid search over the ranges [0.01, 0.1, 1.0, 10.0] and [0.0, 0.2, 0.5, 0.8, 1.0], respectively. The model is optimized using the AdamW (Loshchilov and Hutter 2017) optimizer with hierarchical learning rates: 10^{-4} for the Lora module and 10^{-3} for the other components. The training epoch is set to 3 and the embedding size d is set to 1024. At the end of each epoch, we calculate the model’s loss on the validation set. If the validation loss does not decrease anymore, the model is saved.

Evaluation of Explanation

The text quality and explainability of various explanation generation methods are presented in Table 2. In terms of text

	Yelp		Amazon		TripAdvisor	
	R↓	M↓	R↓	M↓	R↓	M↓
SVD++	1.019	0.791	0.965	0.722	0.809	0.617
DeepCoNN	1.108	0.883	1.108	0.881	0.888	0.683
NARRE	1.031	0.811	1.003	0.780	0.817	0.622
NRT	1.016	0.796	0.954	<u>0.706</u>	0.791	0.605
PETER	<u>1.013</u>	<u>0.783</u>	0.953	0.709	0.806	0.623
CER	<u>1.013</u>	<u>0.787</u>	<u>0.952</u>	0.713	0.814	0.637
CIER	1.009	0.781	0.951	0.705	<u>0.797</u>	<u>0.612</u>

Table 4: The comparison of the recommendation performance of CIER and other baseline methods. “R” means RMSE and “M” means MAE.

quality, our proposed CIER consistently outperforms the baselines on different datasets, demonstrating its effectiveness in generating high-quality sentences. Table 3 presents an example generated by the CIER model and some baselines. By referring to the ground-truth explanation, CIER produces a more accurate explanation.

Regarding explainability, CIER consistently outperforms the baselines on FMR, FCR, and USR, indicating it effectively captures key information in explanatory texts. PEPLER and CIER with vanilla learning, while not explicitly

	GPT-4			Sentiment-Bert			Human annotators		
	Yelp	Amazon	TripAdvisor	Yelp	Amazon	TripAdvisor	Yelp	Amazon	TripAdvisor
PETER	87.2	79.6	87.0	69.2	69.6	80.3	62.0	63.0	84
CER	88.8	80.0	90.4	70.1	70.6	80.7	65.6	66.0	82.5
CIER-M	87.0	81.8	88.0	69.8	74.1	80.1	69.5	60.5	83.5
CIER	90.2	89.8	91.6	70.6	77.6	82.2	73.5	74.0	87.0

Table 5: Results of coherence evaluation using GPT-4, BERT-based sentiment classification models and human annotations for explanations and prediction ratings of the selected methods.

optimized for explainability, demonstrate competitive performance. This can be attributed to their inherent text generation ability obtained by pre-training, enabling them to focus on the nuances and key information within explanations.

Evaluation of Rating Prediction

Evaluation of recommendation accuracy is shown in Table 4. The experimental results indicate that the proposed method, leveraging an LLM backbone, exhibits strong recommendation performance across all datasets, especially excelling in larger datasets (i.e., Yelp and Amazon). In the smaller, sparser TripAdvisor dataset, while traditional models like NRT perform better, our method still outperforms other Transformer-based models (i.e., PETER and CER).

Evaluation of Coherence

The evaluation of the coherence between the explained and predicted ratings is shown in Table 5. The manual annotation was performed by two volunteers who selected 100 data points from each dataset for the selected methods. Before annotation, the agreement between the two instructed annotators was measured using the kappa coefficient on a random sample of 200 data points, resulting in a score of 0.918.

Our approach consistently maintains significant advantages in coherence. In particular, our method consistently outperforms CIER-M, suggesting that our approach of using predicted ratings to guide explanation generation allows the model to understand the relationship between ratings and explanations, thereby improving the relationship between explanations and predicted ratings.

Effect of Keyword Generation Task

To test the effect of our designed keyword generation task, we experimented with three different learning strategies:

- 1) **Vanilla Training**, which involves training solely for rating prediction and explanation generation. While straightforward, it struggles to capture key explanatory words in explanations.
- 2) **Two-Stage Training**, which involves the model first learning to generate keywords before shifting to explanation generation. While this process helps build a solid foundation, it risks the model forgetting keyword generation knowledge.
- 3) **Curriculum Learning (Ours)**, which employs a gradual transition from keyword generation to explanation generation. It reduces the risk of forgetting keyword generation knowledge and minimizes its negative impacts.

	Yelp		Amazon		TripAdvisor	
	FMR \uparrow	R-L \uparrow	FMR \uparrow	R-L \uparrow	FMR \uparrow	R-L \uparrow
CIER	8.71	10.90	12.45	11.70	8.08	13.40
w/o RS	8.66	10.80	12.35	11.65	7.95	13.31
w/o SR2WE	<u>8.67</u>	<u>10.86</u>	<u>12.40</u>	<u>11.68</u>	<u>8.03</u>	<u>13.37</u>
w/o RA	8.58	10.72	12.36	11.59	7.92	13.35

Table 6: Ablation analysis of explanation tasks. “RS” means rating smoothing, “RA” means rating-aware.

All training processes consist of 3 epochs. For Two-Stage Training, the epochs are distributed in a ratio of 1:2 between the first and second stages. The experimental results are shown in Table 2. The two-stage training strategy fails to improve the explainability on the Yelp dataset, possibly due to its large size, which led to knowledge forgetting. Curriculum Learning strategy demonstrates the best performance across all datasets. It makes the model effectively retain and utilize learned knowledge on keyword generation, resulting in more relevant and accurate explanations. However, curriculum learning does not achieve the best performance on the Amazon dataset, likely because 50% of its keywords appear only once, 10% more than that in the other datasets. Thus it is harder to use keywords for generation.

Ablation Study

Table 6 provides the results of the ablation experiments. After removing Rating Smoothing, both the explainability and text quality decline across all datasets.

Moreover, removing the SR2WE and inserting the ratings from rating smoothing directly into the LLM prompts through the linear layer leads to a decrease in model performance, which indicates that the SR2WE module could better embed the ratings into the word vector space.

After disabling Rating-Aware generation, all indicators show a significant decline, indicating that explicit use of rating information is very beneficial for explanation generation.

Conclusion

In this paper, we introduce a novel method that utilizes LLMs as the backbone generation model, predicting ratings and explanations with some tailored training techniques. Additionally, we propose to employ LLMs and pre-trained sentiment analysis models to automatically evaluate the coherency between ratings and explanations. Extensive experimental results demonstrate that our approach outperforms the previous state-of-the-art approaches.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (No. 62072182 and No. 92270119), Shanghai Institute of Artificial Intelligence for Education, and Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education.

References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Al-Taie, M.; and Kadry, S. N. 2014. Visualization of Explanations in Recommender Systems. *Journal of Advanced Management Science*, 2: 140–144.
- Chen, C.; Zhang, M.; Liu, Y.; and Ma, S. 2018. Neural attention rating regression with review-level explanations. In *Proceedings of the 2018 world wide web conference*, 1583–1592.
- Chen, X.; Chen, H.; Xu, H.; Zhang, Y.; Cao, Y.; Qin, Z.; and Zha, H. 2019. Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation. In *SIGIR*, 765–774.
- Cheng, H.; Wang, S.; Lu, W.; Zhang, W.; Zhou, M.; Lu, K.; and Liao, H. 2023. Explainable Recommendation with Personalized Review Retrieval and Aspect Learning. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *ACL*, 51–64.
- Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *EMNLP*, 1724–1734.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *NAACL*, 4171–4186.
- Dong, L.; Huang, S.; Wei, F.; Lapata, M.; Zhou, M.; and Xu, K. 2017. Learning to Generate Product Reviews from Attributes. In Lapata, M.; Blunsom, P.; and Koller, A., eds., *ACL*, 623–632.
- Fu, Z.; Xian, Y.; Gao, R.; Zhao, J.; Huang, Q.; Ge, Y.; Xu, S.; Geng, S.; Shah, C.; Zhang, Y.; and de Melo, G. 2020. Fairness-Aware Explainable Recommendation over Knowledge Graphs. In *SIGIR*, 69–78. New York, NY, USA: Association for Computing Machinery. ISBN 9781450380164.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.
- Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- Hu, S.; Ding, N.; Wang, H.; Liu, Z.; Wang, J.; Li, J.; Wu, W.; and Sun, M. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*.
- Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, 426–434. New York, NY, USA: Association for Computing Machinery. ISBN 9781605581934.
- Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In Knight, K.; Nenkova, A.; and Rambow, O., eds., *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 110–119. San Diego, California: Association for Computational Linguistics.
- Li, L.; Zhang, Y.; and Chen, L. 2020. Generate Neural Template Explanations for Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, 755–764. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368599.
- Li, L.; Zhang, Y.; and Chen, L. 2021. Personalized Transformer for Explainable Recommendation. In *ACL*.
- Li, L.; Zhang, Y.; and Chen, L. 2023. Personalized Prompt Learning for Explainable Recommendation. *ACM Transactions on Information Systems (TOIS)*.
- Li, P.; Wang, Z.; Ren, Z.; Bing, L.; and Lam, W. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. In *SIGIR*, 345–354.
- Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.
- Loshchilov, I.; and Hutter, F. 2017. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.
- Neelakantan, A.; Xu, T.; Puri, R.; Radford, A.; Han, J. M.; Tworek, J.; Yuan, Q.; Tezak, N.; Kim, J. W.; Hallacy, C.; et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.
- Ni, J.; Lipton, Z. C.; Vikram, S.; and McAuley, J. 2017. Estimating Reactions and Recommending Products with Generative Models of Reviews. In Kondrak, G.; and Watanabe, T., eds., *IJCNLP*, 783–791.
- NLP Town. 2023. bert-base-multilingual-uncased-sentiment (Revision edd66ab).
- OpenAI. 2023. GPT-4 Technical Report. *arXiv:2303.08774*.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In Isabelle, P.; Charniak, E.; and Lin, D., eds., *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.
- Raczyński, J.; Lango, M.; and Stefanowski, J. 2023. The Problem of Coherence in Natural Language Explanations of Recommendations. In *ECAI*.
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

- Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. In Erk, K.; and Smith, N. A., eds., *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715–1725. Berlin, Germany: Association for Computational Linguistics.
- Sun, P.; Wu, L.; Zhang, K.; Fu, Y.; Hong, R.; and Wang, M. 2020. Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation. In *WWW*, 837–847.
- Sun, Z.; Shen, Y.; Zhou, Q.; Zhang, H.; Chen, Z.; Cox, D.; Yang, Y.; and Gan, C. 2024. Principle-driven self-alignment of language models from scratch with minimal human supervision. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*. Red Hook, NY, USA: Curran Associates Inc.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, 6000–6010. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.
- Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; and Wei, F. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.
- Yang, A.; Wang, N.; Deng, H.; and Wang, H. 2021. Explanation as a Defense of Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 1029–1037.
- Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X.; et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.
- Zhang, S.; Yao, L.; Sun, A.; and Tay, Y. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1): 1–38.
- Zhang, W.; Yan, J.; Wang, Z.; and Wang, J. 2022. Neuro-Symbolic Interpretable Collaborative Filtering for Attribute-based Recommendation. In *WWW*, 3229–3238.
- Zhang, Y.; and Chen, X. 2020. Explainable Recommendation: A Survey and New Perspectives. *Found. Trends Inf. Retr.*, 1–101.
- Zhang, Y.; Sun, Y.; Zhuang, F.; Zhu, Y.; An, Z.; and Xu, Y. 2023. Triple Dual Learning for Opinion-based Explainable Recommendation. *ACM Transactions on Information Systems*, 42(3): 1–27.
- Zhang, Y.; Zhang, H.; Zhang, M.; Liu, Y.; and Ma, S. 2014. Do users rate or review? Boost phrase-level sentiment labeling with review-level sentiment classification. In *SIGIR*.
- Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the tenth ACM international conference on web search and data mining*, 425–434.
- Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; YU, L.; Zhang, S.; Ghosh, G.; Lewis, M.; Zettlemoyer, L.; and Levy, O. 2023. LIMA: Less Is More for Alignment. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 55006–55021. Curran Associates, Inc.