


Article

Predictive Prompts with Joint Training of Large Language Models for Explainable Recommendation

Ching-Sheng Lin ^{1,*}, Chung-Nan Tsai ², Shao-Tang Su ¹, Jung-Sing Jwo ^{1,3}, Cheng-Hsiung Lee ¹  and Xin Wang ⁴¹ Master Program of Digital Innovation, Tunghai University, Taichung 40704, Taiwan² Lam Research Japan GK, Kanagawa 222-0033, Japan³ Department of Computer Science, Tunghai University, Taichung 40704, Taiwan⁴ Department of Epidemiology and Biostatistics, University at Albany School of Public Health, State University of New York, Rensselaer, NY 12144, USA

* Correspondence: cslin612@thu.edu.tw

Abstract: Large language models have recently gained popularity in various applications due to their ability to generate natural text for complex tasks. Recommendation systems, one of the frequently studied research topics, can be further improved using the capabilities of large language models to track and understand user behaviors and preferences. In this research, we aim to build reliable and transparent recommendation system by generating human-readable explanations to help users obtain better insights into the recommended items and gain more trust. We propose a learning scheme to jointly train the rating prediction task and explanation generation task. The rating prediction task learns the predictive representation from the input of user and item vectors. Subsequently, inspired by the recent success of prompt engineering, these predictive representations are served as predictive prompts, which are soft embeddings, to elicit and steer any knowledge behind language models for the explanation generation task. Empirical studies show that the proposed approach achieves competitive results compared with other existing baselines on the public English TripAdvisor dataset of explainable recommendations.

Keywords: large language models; recommendation systems; human-readable explanations; rating prediction task; explanation generation task; prompt engineering; predictive prompt

MSC: 94A16



Citation: Lin, C.-S.; Tsai, C.-N.; Su, S.-T.; Jwo, J.-S.; Lee, C.-H.; Wang, X. Predictive Prompts with Joint Training of Large Language Models for Explainable Recommendation. *Mathematics* **2023**, *11*, 4230. <https://doi.org/10.3390/math11204230>

Academic Editors: Adrian Sergiu Darabant and Diana-Laura Borza

Received: 23 August 2023

Revised: 22 September 2023

Accepted: 4 October 2023

Published: 10 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recommendation systems have been widely adapted in various applications including e-commerce sites, social media usages, and content platforms to address the information overload [1]. The recommendation engines help in predicting the right items to the customer based on the histories of other similar users, the attributes of users and items, and the customer's explicit or implicit personal preferences. There are two major approaches to solve this research problem where Collaborative Filtering (CF) assumes people who share similar past preferences will likely have similar preferences in the future and Content-Based Filtering (CBF) suggests items to users based on the similarity comparison between users' preferences and items' characteristics [2].

However, traditional recommendation techniques have less interaction with users to discover their real-time requirements, resulting in inaccurate predictions and causing a bad user experience. To overcome these challenges, conversational recommendation systems have emerged as a promising alternative which enables users to interact with the recommendation agents through natural language dialogue. This natural and intuitive interaction mode not only provides an avenue to better understand customers' needs but also offers a better user experience [3]. With the recent advancement and popularity of conversational agents (especially the game-changer, ChatGPT), conversational AIs are

able to communicate with users on a broad range of topics like never before [4]. Figure 1 demonstrates a conversational recommendation system between a user and an agent for multiple turns shown in black text. Although a massive amount of recommendation algorithms are powerful enough to generate accurate recommendations, most of them fail to produce explanations for their recommendations and lack transparency.

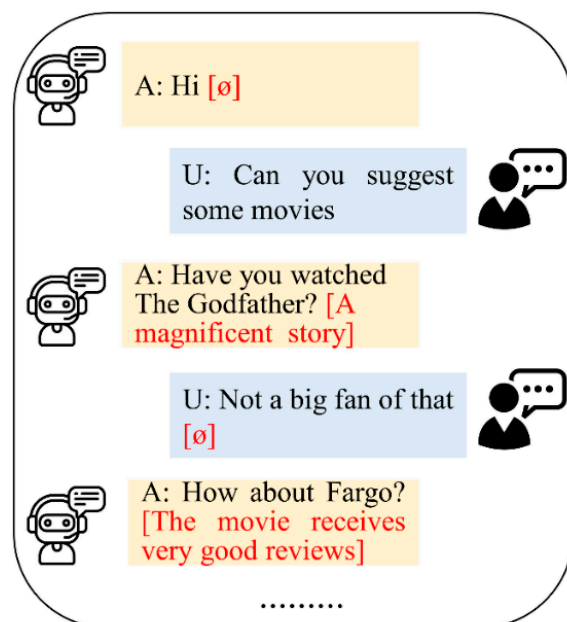


Figure 1. A dialogue example of an explainable recommendation system.

There is a general agreement that both accuracy and explainability are crucial aspects for the evaluation of recommendation systems [5,6]. Applying text explanations to the recommended items would increase user trust as well as help users easily make decisions and boost user satisfaction. Hence, more and more research efforts have been devoted to improving the recommendation and enhancing the explanations for the non-explainable black box recommender systems [7]. The red text in Figure 1 illustrates the explanations of the recommendation system.

Most recently, large-scale pre-trained language models (PLMs) have been the focus of mainstream media and achieved outstanding performance in various natural language processing tasks [8]. The main reason that PLMs have taken the Internet by storm is due to their easy adoption. GPT series models trained to predict the next word in an autoregressive process have gained the most attention and instigated a training paradigm shift from pre-training and fine-tuning to prompt learning [9].

Since PLMs are usually trained on large corpora and are difficult to optimize for small training data, it is important to develop an approach to efficiently exploit and influence PLMs' output. Prompt-based learning is an emerging solution by automatically generating prompts in order to adopt PLMs to perform downstream tasks [10]. In this paper, we take user id and item id as the input to generate explanations for the recommendations by leveraging the PLMs. The proposed learning scheme jointly trains the rating prediction task and explanation generation task. The rating prediction task learns the predictive representation from the input of user and item vectors. Subsequently, these predictive representations are served as predictive prompts, which are soft embeddings, to elicit and steer any knowledge behind language models for the explanation generation task. Compared with the hard prompts which have to rely on human experts to design templates and are difficult to optimize, soft prompts could be directly fine-tuned using data from downstream tasks and possess more representation ability. Unlike the traditional joint training strategy which mainly optimizes multiple tasks simultaneously, our end-to-end

joint model also explicitly incorporates features extracted from one task into the other task for the purpose of the interaction between tasks.

The key contributions of our proposed approach are two-fold:

- We propose a joint training scheme to predict the recommendation rating and produce explanations of the recommendation based on the prompt learning. The predictive prompts are taken from the predictive representations learned in the rating prediction task, and fed into PLMs to generate output text.
- Experiments are conducted on the TripAdvisor dataset to verify the effectiveness of our approach on both rating prediction task and explanation generation task. The results show that our method is not only capable of generating suitable explanations but also achieves promising performance comparable with other state-of-the-art algorithms in terms of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for the rating prediction task.

The rest of this paper is structured as follows. In Section 2, several research fields and paradigms related to the works of this paper are reviewed. In Section 3, an in-depth discussion of our joint model is provided and discussed. In Section 4, empirical studies are performed on the public dataset to examine the effectiveness and further analyze the results of the proposed model. In the last section, we make some conclusions and briefly present some possible future investigations.

2. Related Work

Since our proposed model is a joint training framework to predict the recommendation rating and generate explanations of the recommendation based on the predictive prompts using language models, we review three related research directions including the recommendation system, pre-trained language models and prompt learning in this section.

2.1. Recommendation Systems

Since the earliest recommendation algorithm in the 1990s, this research field has always been actively investigated by both industry and academia [11]. Content-based filtering recommendation algorithms have improved their performance by operating on embedding representations compared to the traditional feature-engineering approaches such as Bag-of-Words and TF-IDF [12]. Collaborative filtering is another popular technique in recommender systems to predict user interests by analyzing the behaviors and opinions of similar users. Since graph neural networks (GNNs) have facilitated the representation learning in recent years, the Graph Matching-based Collaborative Filtering model (GMCF) considers two types of attribute interactions where inner interactions independently operate on either user-specific or item-specific attributes, and cross interactions further integrate the interdependent relationships between user-specific and item-specific attributes [13]. A recommendation system based on sentiment analysis and matrix factorization (SAMF) was proposed to address data sparsity and credibility. It combines topic models, matrix factorization and deep learning to enhance recommendation accuracy [14].

Compared to traditional recommender systems, the conversational recommendation system provides a human-like interaction that enables users to receive personalized recommendations which are highly relevant to their needs and truly beneficial for their decision making [15]. To address the cold-start users and static model issues, the ConTS model is designed to dynamically ask users to provide their favorite attributes of the item and then make rational recommendations [16]. The EAR model is proposed to estimate the user preference over items (Estimation stage), learn a reinforced dialogue policy on multi-round conversational recommendations (Action stage), and update the learning model based on users' feedback (Reflection stage) [17].

As the explanation in the recommendation systems is becoming more and more important, a Deep Explicit Attentive Multi-View Learning Model (DEAML) adopts a knowledge graph to produce explanations, and applies an attentive multi-view learning model for solving the rating prediction problem [5]. To leverage the user interaction in the

recommendation system, a multitask learning framework is used to simultaneously train the recommendation task, explanation generation, and user feedback incorporation [6].

2.2. Pre-Trained Language Models

The Transformer architecture is an encoder–decoder structure with self-attention mechanism to learn long-range representation that was initially applied to natural language processing tasks and has been widely extended to various domains [18]. It often follows the self-supervised pre-training strategy and enables fast adaptation to downstream tasks. Based on the pre-training mechanism, there are three variants of structures including Transformer encoders, Transformer decoders and full Transformers with encoder–decoders [19].

The objective of the Transformer encoder framework aims to predict hidden text using a bidirectional Transformer encoder. BERT which is pre-trained on Masked Language Model (MLM) and Next Sentence Prediction (NSP) is one of the most remarkable models in this category that achieves state-of-the-art performances on many NLP tasks [20]. The Transformer decoder framework employs an autoregressive generation by predicting the next text conditioned on the past sequence. GPT series models apply left-to-right unidirectional modeling to pre-train on large scale corpus and allow in-context learning without fine-tuning the original model parameters [21,22]. The Transformer encoder–decoder architecture is a sequence-to-sequence learning model where the encoder masks several fragments and the decoder manages to recover these hidden sections. BookGPT guides language models to predict user ratings textually for book recommendation tasks where the output format is based on predefined templates [23]. Generative recommendation (GenRec) is a recommendation system that leverages large language models to directly generate recommended items, instead of using traditional ranking-based approaches by calculating the rating score for each candidate item [24].

2.3. Prompt Learning

Prompt-based learning aims at learning or developing suitable and effective representations that can elicit the large PLMs to carry out the given task or solve the specific problem. This new paradigm of learning has achieved a lot of success in diverse NLP tasks because of its adaptability and capability.

Automated prompt engineering can be classified into discrete prompts and continuous prompts [10]. Discrete prompts work on searching natural language text and appending these text as prompts to reformulate the downstream task. For example, to perform the sentiment analysis of the given sentence “I like the game very much”, a possible discrete prompt could be like “The game is” and the PLMs would be leveraged to predict a next token for determining the sentiment of the sentence [25]. LAPAQA generates discrete prompts by extracting the text located between the input and output from large text corpora [26]. Continuous prompts, in contrast to the discrete prompting method which requires human-understandable text, enable the model to learn from the continuous vector space of the underlying PLMs. Prefix-Tuning is a novel technique to guide the PLMs towards the designated task where task-specific vectors are prepended to the input sequence as prefixes [9]. In the prompt-based news recommendation (PBNR) approach, news recommendation is considered as a text-to-text language task, taking into account users’ past reading behaviors. During model training, both ranking loss and language generation loss are integrated [27]. A Prompt Learning for News Recommendation (Prompt4NR) framework changes the prediction task from determining if a user would click on a candidate news article into a cloze-style mask-prediction task with various prompt templates [28].

3. Proposed Method

The objective of this paper is to recommend item i to user u and provide a comprehensive explanation simultaneously. Our network model to address the explainable recommendation problem is a joint training framework based on prompt-based learning for the large language model and is depicted in Figure 2. There are two major components

in our system. The first component is a deep neural network used to predict a recommendation score from a given user–item pair (left-hand side of Figure 2). The second component utilizes the latent representation learned from the first component as the predictive prompts, feeding them into a large PLM to generate explanations (right-hand side of Figure 2).

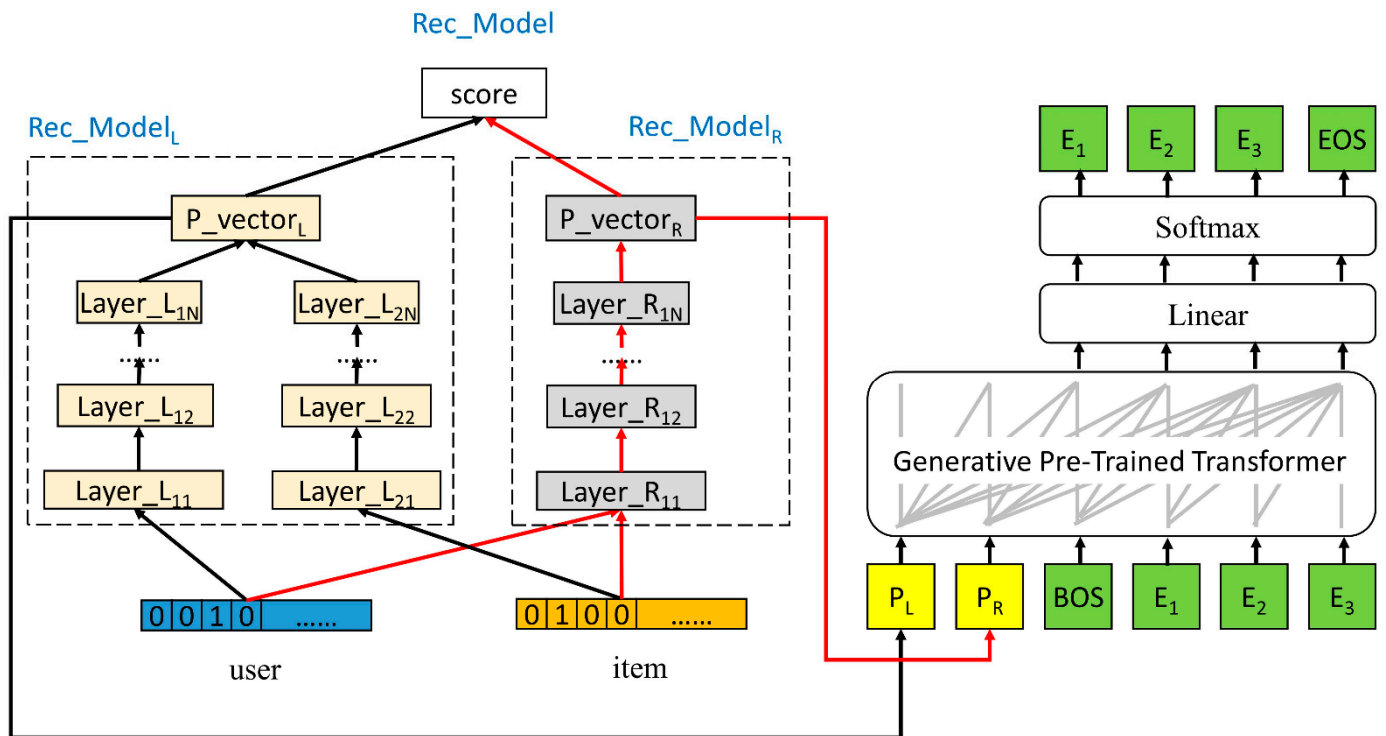


Figure 2. The architecture of prompt-based explainable recommendation model.

3.1. Prompt-Based Explainable Recommendation Architecture

In this research, we address the problem of explainable recommendation by proposing a joint model for recommendation score prediction and explainable text generation, exploiting the dependencies between these two tasks to enhance the performance compared to independent models. Joint training is a frequently employed approach when closely related tasks can be trained simultaneously and share similar datasets. In the context of neural network models, as these tasks share specific model layers, it promotes the model to acquire more universally applicable representations, thereby maximizing its generalization performance across all tasks. The recommendation score prediction is based on Collaborative Filtering of fusion learning [29]. Once the score has been established, the predictive vectors obtained during the learning process are leveraged with the use of pre-trained language models, which provide the rich information derived from large training corpus, to generate explanations.

Inspired from the DeepCF [29], our recommendation model (Rec_Model) contains two branches of deep network where one is Rec_Model_L to learn the representation and the other is Rec_Model_R to learn the match function (left-hand side of Figure 2). Unlike DeepCF which takes user ratings and item ratings as inputs, Rec_Model directly uses IDs of users and items. The Rec_Model_L consists of two subnetworks represented by $Layer_{L_{1n}}$ and $Layer_{L_{2n}}$, respectively. The input of $Layer_{L_{11}}$ is the user id and the input of $Layer_{L_{21}}$ is the item id. By combining the last layers $Layer_{L_{1N}}$ and $Layer_{L_{2N}}$, the final representation is P_vector_L . Regarding the structure of Rec_Model_R , the concatenation of the user id and

item id is passed into the Layer_{R11} and is processed through multiple layers to form the final representation P_vector_R. The Rec_Model_L can be formulated as:

$$\begin{aligned} X_j^{L1} &= a\left(X_{j-1}^{L1} W^{L1}\right) \\ X_j^{L2} &= a\left(X_{j-1}^{L2} W^{L2}\right) \\ P_vector_L &= X^{L1} \odot X^{L2} \end{aligned} \tag{1}$$

where W^{L1} and X_j^{L1} define the weight matrix and input of the j-th layer, respectively. $a(\cdot)$ denotes the activation function to learn more complex representation. The same notations are applied to the subnetwork Layer_{L2n} as well. Subsequently, the element-wise product is employed to the outputs of two subnetworks to yield P_vector_L. Meanwhile, the formulation of Rec_Model_R is described below:

$$\begin{aligned} X_j^{R1} &= a\left(X_{j-1}^{R1} W^{R1}\right) \\ P_vector_R &= X^{R1} \end{aligned} \tag{2}$$

where W^{R1} and X_j^{R1} are the weight matrix and input of the j-th layer. The final X^{R1} is symbolized by P_vector_R. Given the outputs of two branches, we combine them and make the prediction score as follows:

$$\hat{s}_{u,i} = a\left(W^F \begin{bmatrix} P_vector_L \\ P_vector_R \end{bmatrix}\right) \tag{3}$$

where W^F is the weight matrix and $\hat{s}_{u,i}$ is the prediction score of the recommendation.

After the recommendation process has finished, we propose a prompt learning method to elicit the knowledge behind the pre-trained language models for the explanation generation task (right-hand side of Figure 2). In this part, P_vector_L and P_vector_R are served as predictive prompts represented by P_L and P_R to trigger the explanation words and GPT-2 is used as the pre-trained language generation model. We made extensive use of the GPT2 libraries created by the HuggingFace community. These libraries provide a wide range of robust deep learning tools and efficiently integrate with the PyTorch framework. GPT2 has been trained on large amount of data and significantly demonstrated remarkable performance across many natural language processing tasks. During the training phase, the prompts with explanation words (e_j) are fed into GPT-2 to obtain the encoding X^E . Then, X^E is passed through a linear layer with softmax activation to generate a probability distribution over the full vocabulary. The formulation can be expressed as

$$d_j = \text{softmax}\left(X^E W^E\right) \tag{4}$$

where d_j is the probability distribution and W^E is the weight matrix.

3.2. Learning Process

During the training stage, to learn the recommendation score prediction module, we set the loss function using mean square error as:

$$\text{Loss}_R = \frac{1}{|\text{Tr}|} \sum (\hat{s}_{u,i} - s_{u,i})^2 \tag{5}$$

where $s_{u,i}$ is the ground-truth recommendation score and $|\text{Tr}|$ denotes the size of training samples. To learn the explanation generation, we choose negative log-likelihood as the loss function, which is defined below:

$$\text{Loss}_E = \frac{1}{|\text{Tr}|} \sum \frac{1}{|\text{Exp}|} \sum -\log d_j^{e_j} \tag{6}$$

where $|\text{Exp}|$ denotes the number of explanation words. To jointly model both recommendation score prediction and explanations, we combine two tasks into a multi-task learning architecture and formulate the loss function as:

$$\text{Loss}_T = \min_{\theta} (\text{Loss}_R + \alpha \text{Loss}_E) \quad (7)$$

where α is the weighted parameter to balance two terms.

In the inference phase, initially, we use the predictive prompts with a special word BOS as the input. The autoregressive process takes the input, runs through the explanation generation network, selects a word based on the generated word probability distribution over the vocabulary and appends the selected word to the end of the input to form the new input. Then, the autoregressive process is repeated until a special generated word EOS is produced [30].

Algorithm 1 demonstrates the training procedure of our proposed model.

Algorithm 1: Prompt-Based Explainable Recommendation Model.

Input: The training dataset D {User (U), Item (I), Explanation (E)}

Output: θ for all the trainable weights in the model

```

1: Randomly initialize  $\theta$ 
2: repeat:
3:   for each mini-batch  $\{u, i, e\}$  in  $D$ :
4:     Calculate  $P_{\text{vector}_L}$  and  $P_{\text{vector}_R}$  to obtain  $\hat{s}_{u,i}$ 
5:     Compute the  $\text{Loss}_R$  in Equation (5)
6:     Calculate  $d_j$  based on the expression of Equation (4)
7:     Compute the  $\text{Loss}_E$  in Equation (6)
8:     Compute the  $\text{Loss}_T$  in Equation (7)
9:     Update the weight  $\theta$  to minimize the  $\text{Loss}_T$ 
10:   end for
11: until convergence:

```

4. Experiments and Results

In this section, we develop empirical studies to assess the proposed method in recommendation and explanation tasks. The flow of the experiments includes data-processing, training, testing and ablation studies. The data are described and divided into different sets for the evaluation (Section 4.1). We train the model for baselines and our approach with training set, subsequently evaluating the performance of each model in terms of the evaluation criterion (Section 4.2) with separate test data after the training procedure (Section 4.3). The ablation studies are conducted to examine the impact and importance of each component in our model (Section 4.4).

4.1. Dataset

The source used for evaluating our model is the TripAdvisor dataset which contains 9765 users, 6280 items and 320,023 reviews. The ground-truth explanations are those sentences from reviews. Similar to the prior research settings [31,32], the training, validation, and testing sets are divided in an 8:1:1 ratio, resulting in 256,017 training samples, 32,003 validation samples and 32,003 testing samples, respectively. Sample data can be seen in Figure 3.

```

{'user': 1, 'item': 0, 'rating': 5, 'text': 'we had to wait for the room to be ready and we good a free up grade as a compensation'},
{'user': 53, 'item': 0, 'rating': 4, 'text': 'the pool was really high up and has an amazing view'},
{'user': 56, 'item': 0, 'rating': 5, 'text': 'the pool is just fabulous'},
{'user': 58, 'item': 0, 'rating': 5, 'text': 'the gym is set up high to give you stunning views of hong kong harbour while you work'},

```

Figure 3. Sample data.

4.2. Evaluation Criterion

To assess the overall performance of our proposed solution automatically, both the recommendation task and explanation generation need to be evaluated. Regarding the measurement of recommendation module, we employ two popular metrics, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) [11]. For the generation of explanations, we adopt BLEU (Bilingual Evaluation Understudy) [33] and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [34] as two evaluation indicators.

ROUGE-N Precision calculates the ratio of the number of N-grams in the system results that also appear in the reference results, divided by the total number of N-grams in the system results. ROUGE-N Recall measures the proportion of N-grams in the reference results that are also found in the system results, divided by the total number of N-grams in the reference results. ROUGE-N F-score is the harmonic mean of ROUGE-N Precision and ROUGE-N Recall. In this research, we report ROUGE-1 F-score (ROUGE1-F) and ROUGE-2 F-score (ROUGE2-F). The BLEU score automatically evaluates sentence generation quality by calculating n-gram precision between system outputs and reference results, while also incorporating a brevity penalty (BP) to avoid excessively brief sentences. We use Bleu-1 and Bleu-4 scores to assess the quality and fluency of generated explanations. It is worth to note that although BLEU and ROUGE scores are typical and standard metrics for evaluation, focusing only on the lexical overlap between system-generated and human-written answers may be biased. Adapting these two metrics to better correlate lexical overlap with the human judgment is an important future research direction.

4.3. Experimental Performances

To validate the efficacy, our proposed method is compared with several state-of-the-art models as the baselines:

- Att2Seq [35]: This method is an attention-enhanced attribute-to-sequence network and is initially proposed to create product reviews. The model uses the user ID, product ID, and rating as attributes.
- NRT [36]: Unlike reviews which are lengthy and time consuming, tips are very succinct insights to capture user experience with only a few words in E-commerce sites. This paper uses multi-task learning framework to predict product ratings and generate tips where the rating prediction is based on a multi-layer perceptron network and tip generation is a sequence decoder model. The input of this model consists of user id and item id.
- PEPLER-MF [32]: PEPLER is a personalized prompt-based learning for explainable recommendation using pre-trained language models based on user and item IDs. PEPLER-MF uses Matrix Factorization (MF) for the recommendation rating score prediction by the user and item embeddings. The explanations are produced with the aid of pre-trained language models.
- PEPLER-MLP [32]: This method is another variant of PEPLER. The major difference is that this model trains a Multi-Layer Perceptron (MLP) to estimate the rating scores.

The experimental comparison is operated in the environment of a Windows 10 OS, utilizing an Intel Core i9 central processing unit (CPU) with 128 GB of memory, and a GPU NVIDIA GeForce RTX 3090 with a memory size of 24 GB. We adjust the hyper-parameters empirically to ensure the training quality and the hyper-parameter settings can be seen in Table 1.

According to the comparison results displayed in Table 2 where the first two baselines (Att2Seq and NRT) are directly obtained from the relevant papers and the results of the rest (PEPLER-MF and PEPLER-MLP) are generated by the provided source codes of those papers, we achieve the highest scores with respect to BLEU-1, ROUGE1-F, ROUGE2-F and MAE, demonstrating the capability to make appropriate recommendations and create acceptable explanations. Regarding the performance of recommendation task measured in RMSE and MAE, in addition to PEPLER-MF, most methods exhibit similar performances aligned with the prior research findings [37]. Our method receives high scores on BLEU-1

but lacks consistent improvements on higher-order BLEU and ROUGE scores. Since GPT2 is a generation-based language model, it may produce synonyms or paraphrases of reference text, which have similar or identical semantic content but receive lower ROUGE and BLEU scores. Combining automated metrics and human evaluation to assess the performance of a generation-based language model comprehensively is very important and is worth further investigation.

Table 1. Hyper-parameter configuration.

Parameter	Value
batch size	128
epoch	10
P_vector _L size	768
P_vector _R size	768
layers N in Rec_Model	2
learning rate	0.00075

Table 2. Comparative assessment of all competing methods on TripAdvisor dataset.

	BLEU-1	BLEU-4	ROUGE1-F	ROUGE2-F	RMSE	MAE
Att2Seq	15.20%	0.96%	16.38%	2.19%	-	-
NRT	13.76%	0.80%	15.58%	1.68%	0.790	0.610
PEPLER-MF	15.94%	1.14%	16.38%	2.14%	1.574	1.341
PEPLER-MLP	15.91%	1.03%	16.39%	2.13%	0.799	0.612
Our Model	16.45%	1.10%	16.71%	2.24%	0.795	0.607

In Table 3, we additionally list various examples, including the ground truth (Ground truth_X) and the generated explanations (Generation_X), to demonstrate our results. From what we can see, the polarity of each generated explanation aligns with the ground truth. However, there is still significant potential for improvement in context generation. As observed in example 5, although the sentiment of the explanation is correct, the subjects are not accurately described. Incorporating more information of users and items could be an avenue to direct the language model's generation.

Table 3. Five example outputs produced by our model and the corresponding ground truths.

Ground truth_1: location was good and was close to many restaurants
Generation_1: the hotel is located in a great location
Ground truth_2: pool area is good though
Generation_2: the pool is great and the staff are very helpful
Ground truth_3: the bed is very comfortable and the bath room is great
Generation_3: the bed was very comfortable and the bathroom was clean and modern
Ground truth_4: i enjoyed the front desk staff
Generation_4: the front desk staff was very helpful

Table 3. *Cont.*

Ground truth_5: gym also very small and with an odd smell
Generation_5 the room was very small and the bathroom was very small

4.4. Ablation Studies

To validate the applicability of our suggestions, we execute ablation studies on the TripAdvisor dataset to explore the advantages of our methodology and quantify the impact of each critical module. Referring to the findings presented in Table 4, removing either P_vector_L or P_vector_R leads to performance degradation in terms of all evaluation metrics. Hence, we are confident that our approach can significantly improve the performance of explainable recommendation by jointly learning the recommendation module and explanation generation.

Table 4. Ablation experimental results.

	BLEU-1	BLEU-4	ROUGE1-F	ROUGE2-F	RMSE	MAE
Our Model	16.45%	1.10%	16.71%	2.24%	0.795	0.607
- P_vector_L	15.89%	1.03%	16.43%	2.14%	0.796	0.611
- P_vector_R	15.81%	1.03%	16.41%	2.17%	0.798	0.612

5. Conclusions

In this work, we propose a prompt-based method to address the explainable recommendation system with the leverage of pre-trained language models. To optimize both recommendation correctness and explanation generation simultaneously, we apply a multi-task learning strategy. We perform experimental studies on the TripAdvisor dataset and yield satisfactory results in terms of BLEU-1, ROUGE1-F, ROUGE2-F and MAE. The direction of adopting predictive prompts in pre-trained language models provides a promising alternative to produce reasonable recommendation explanations.

Our future work is to expand upon this paper by focusing on the following areas. First, as the current approach only uses the embeddings learned from the rating task as predictive prompts, we will investigate more predictive prompts for explanation generation such as the item descriptions and users' preferences. Second, the research of recommendation systems is still an evolving field [38] and we intend to delve further into more advanced recommendation models. With the increasing quality of recommendation accuracy, we expect to enhance both recommendation performance and explanation generation at once. Third, the current language model is based on GPT-2 and we will explore other knowledge enhanced pre-trained language models to improve the explanation generation [39].

Author Contributions: Supervision, J.-S.J.; methodology, C.-S.L. and C.-H.L.; investigation, C.-S.L., C.-N.T. and S.-T.S.; writing—review and editing, C.-S.L. and X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is financially supported by the National Science and Technology Council (NSTC) of Taiwan under Grant 112-2221-E-029 -019 -.

Data Availability Statement: TripAdvisor datasets can be found <https://github.com/lileispices/PEPLER> (accessed on 13 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, P.; Thakur, R.S. Recommendation system techniques and related issues: A survey. *Int. J. Inf. Technol.* **2018**, *10*, 495–501. [[CrossRef](#)]
2. Barkan, O.; Koenigstein, N.; Yogeve, E.; Katz, O. CB2CF: A neural multiview content-to-collaborative filtering model for completely cold item recommendations. In Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen Denmark, 16–20 September 2019; pp. 228–236.
3. Liu, Z.; Wang, H.; Niu, Z.; Wu, H.; Che, W.; Liu, T. Towards Conversational Recommendation over Multi-Type Dialogs. In Proceedings of the 58th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Online, 5–10 July 2020; pp. 1036–1049.
4. Van Dis, E.A.; Bollen, J.; Zuidema, W.; van Rooij, R.; Bockting, C.L. ChatGPT: Five priorities for research. *Nature* **2023**, *614*, 224–226. [[CrossRef](#)] [[PubMed](#)]
5. Gao, J.; Wang, X.; Wang, Y.; Xie, X. Explainable recommendation through attentive multi-view learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3622–3629.
6. Chen, Z.; Wang, X.; Xie, X.; Parsana, M.; Soni, A.; Ao, X.; Chen, E. Towards explainable conversational recommendation. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 2994–3000.
7. Alshammari, M.; Nasraoui, O.; Sanders, S. Mining semantic knowledge graphs to add explainability to black box recommender systems. *IEEE Access* **2019**, *7*, 110563–110579. [[CrossRef](#)]
8. Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Zettlemoyer, L.; et al. Opt: Open pre-trained transformer language models. *arXiv* **2022**, arXiv:2205.01068.
9. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; Volume 1, pp. 4582–4597.
10. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [[CrossRef](#)]
11. Ko, H.; Lee, S.; Park, Y.; Choi, A. A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics* **2022**, *11*, 141. [[CrossRef](#)]
12. Chen, K.; Liang, B.; Ma, X.; Gu, M. Learning audio embeddings with user listening data for content-based music recommendation. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, Canada, 6–11 June 2021; pp. 3015–3019.
13. Su, Y.; Zhang, R.; MErfani, S.; Gan, J. Neural graph matching based collaborative filtering. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Online, 11–15 July 2021; pp. 849–858.
14. Liu, N.; Zhao, J. Recommendation system based on deep sentiment analysis and matrix factorization. *IEEE Access* **2023**, *11*, 16994–17001. [[CrossRef](#)]
15. Radlinski, F.; Boutilier, C.; Ramachandran, D.; Vendrov, I. Subjective attributes in conversational recommendation systems: Challenges and opportunities. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 12287–12293.
16. Li, S.; Lei, W.; Wu, Q.; He, X.; Jiang, P.; Chua, T.S. Seamlessly unifying attributes and items: Conversational recommendation for cold-start users. *ACM Trans. Inf. Syst.* **2021**, *39*, 1–29. [[CrossRef](#)]
17. Lei, W.; He, X.; Miao, Y.; Wu, Q.; Hong, R.; Kan, M.Y.; Chua, T.S. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In Proceedings of the 13th International Conference on Web Search and Data Mining, Online, 10–13 July 2020; pp. 304–312.
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [[CrossRef](#)]
19. Wang, H.; Li, J.; Wu, H.; Hovy, E.; Sun, Y. Pre-Trained Language Models and Their Applications. *Engineering* **2022**, *25*, 51–65. [[CrossRef](#)]
20. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
21. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
22. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
23. Zhiyuli, A.; Chen, Y.; Zhang, X.; Liang, X. BookGPT: A General Framework for Book Recommendation Empowered by Large Language Model. *arXiv* **2023**, arXiv:2305.15673.
24. Ji, J.; Li, Z.; Xu, S.; Hua, W.; Ge, Y.; Tan, J.; Zhang, Y. Genrec: Large language model for generative recommendation. *arXiv* **2023**, arXiv:2307.
25. Cai, X.; Xu, H.; Xu, S.; Zhang, Y. BadPrompt: Backdoor Attacks on Continuous Prompts. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 37068–37080.

26. Jiang, Z.; Xu, F.F.; Araki, J.; Neubig, G. How can we know what language models know? *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 423–438. [[CrossRef](#)]
27. Li, X.; Zhang, Y.; Malthouse, E.C. PBNR: Prompt-based News Recommender System. *arXiv* **2023**, arXiv:2304.07862.
28. Zhang, Z.; Wang, B. Prompt learning for news recommendation. *arXiv* **2023**, arXiv:2304.05263.
29. Deng, Z.H.; Huang, L.; Wang, C.D.; Lai, J.H.; Philip, S.Y. Deepcf: A unified framework of representation learning and matching function learning in recommender system. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 61–68.
30. He, T.; Tan, X.; Xia, Y.; He, D.; Qin, T.; Chen, Z.; Liu, T.Y. Layer-wise coordination between encoder and decoder for neural machine translation. *Adv. Neural Inf. Process. Syst.* **2018**, *31*. Available online: <https://api.semanticscholar.org/CorpusID:54088698> (accessed on 9 October 2023).
31. Geng, S.; Fu, Z.; Ge, Y.; Li, L.; De Melo, G.; Zhang, Y. Improving personalized explanation generation through visualization. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Volume 1, pp. 244–255.
32. Li, L.; Zhang, Y.; Chen, L. Personalized prompt learning for explainable recommendation. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–26. [[CrossRef](#)]
33. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
34. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
35. Dong, L.; Huang, S.; Wei, F.; Lapata, M.; Zhou, M.; Xu, K. Learning to generate product reviews from attributes. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; Volume 1, pp. 623–632.
36. Li, P.; Wang, Z.; Ren, Z.; Bing, L.; Lam, W. Neural rating regression with abstractive tips generation for recommendation. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 345–354.
37. Li, L.; Zhang, Y.; Chen, L. Personalized Transformer for Explainable Recommendation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Bangkok, Thailand, 1–6 August 2021; Volume 1.
38. Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Trans. Recomm. Syst.* **2023**, *1*, 1–51. [[CrossRef](#)]
39. Hu, L.; Liu, Z.; Zhao, Z.; Hou, L.; Nie, L.; Li, J. A Survey of Knowledge Enhanced Pre-Trained Language Models. *IEEE Trans. Knowl. Data Eng.* **2023**, 1–19. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.