A Unified Prompt-aware Framework for Personalized Search and Explanation Generation

HAOBO ZHANG, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
QIANNAN ZHU , School of Artificial Intelligence, Beijing Normal University, Beijing, China, and Engineering Research Center of Intelligent Technology and Educational Application, MOE, China
ZHICHENG DOU , Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

Product search is crucial for users to find and purchase products they need. Personalized product search, which models users' search intent and provides tailored results, has become a prominent research problem in industry and academia. Recent studies often leverage knowledge graphs (KGs) to improve search performance and generate explanations for search results. However, existing KG-based methods treat search and explanation tasks separately and explore paths in KGs as explanations, creating a gap between search results and generated explanations. Also, path-formed explanations in KGs are not flexible enough to build correlations with the user's current query. To address these challenges, we propose P-PEG, a unified prompt-aware framework for personalized product search and explanation generation. P-PEG leverages a pre-trained language model (PLM) and search signal to enhance the generation of user-understandable explanations. We introduce a prompt learning technique and design prompt generators for search and explanation generation tasks based on a fixed PLM. By incorporating search results in explanation-based prompts, we bridge the gap between search results and explanations, facilitating better interaction. Additionally, we utilize the user's current query, historical search log, and KGs to personalize the explanations and inject task knowledge into PLM. Experimental results show that P-PEG outperforms existing methods in the explanation generation task of the three datasets and the search task of the Electronics dataset, and achieves comparable performance in the search task of the Cellphones & Accessories and CD & Vinyl datasets.

CCS Concepts: • **Information systems** → **Retrieval models and ranking**;

Additional Key Words and Phrases: Explanation generation, Product search, Prompt learning

ACM Reference format:
Haobo Zhang, Qiannan Zhu, and Zhicheng Dou . 2025. A Unified Prompt-aware Framework for Personalized Search and Explanation Generation. *ACM Trans. Inf. Syst.* 43, 3, Article 71 (March 2025), 26 pages.
https://doi.org/10.1145/3716131

This work was supported by National Science and Technology Major Project (No. 2022ZD0120103), National Natural Science Foundation of China (Nos. 62272467, 62472038, and 62437001), Fundamental Research Funds for the Central Universities (No. 2233100004), and the fund for building world-class universities (disciplines) of Renmin University of China.

Authors' Contact Information: Haobo Zhang, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, Beijing, China; e-mail: zhanghb@ruc.edu.cn; Qiannan Zhu (corresponding author), School of Artificial Intelligence, Beijing Normal University, Beijing, China, and Engineering Research Center of Intelligent Technology and Educational Application, MOE, China; e-mail: zhuqiannan@bnu.edu.cn; Zhicheng Dou (corresponding author), Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China; e-mail: dou@ruc.edu.cn.

ACM 1558-2868/2025/3-ART71
https://doi.org/10.1145/3716131

1 Introduction

Product search systems are widely used in various web applications and services as they efficiently connect users with the products they are likely to be interested in, ensuring effective distribution of online content. With the exponential growth of online marketplaces, efficient and accurate product search methods have become increasingly critical. In the typical product search scenario, the users interact with the product search engine by submitting queries, and in return, they receive a list of products ranked based on the search methods that estimate the likelihood of purchase. By enhancing the accuracy of the search methods in ranking products, users can experience a faster and more convenient selection of their desired products, contributing to increasing platform turnover.

Because users have different interests in various aspects of products, in recent years some researchers developed personalized product search methods to achieve higher search accuracy. These personalized methods mainly make use of the user's historical search logs to model users' personalized interests and preferences, so that the search results under the user's current query are more in line with users' preferences for goods. Some earlier studies like HEM [3], ALSTP [13] and ZAM [1] apply the language model on user's historical reviews to build user preference for retrieving products of interest to the user. Some researchers after that try to apply the transformer to personalized product search [5, 49]. By using the transformer, they can process the user's search sequences more accurately to extract the user's interests and estimate the purchasing likelihood of the products under the current query. Although these personalized methods have been proven to significantly enhance search results in terms of accuracy, they have the problem of not being able to explain to users why these products are retrieved. The absence of a solid explanation for search results may leave users perplexed and further affect users' trust in the E-commerce platforms.

Inspired by the successful use of KGs in the explainable product recommendations [25, 35], many personalized product search methods attempt to leverage KGs to generate path-formed explanations and achieve the most promising search performance. For example, DREM [4] constructs a user–product KG and regards the queries as dynamic relations between users and products. It then leverages an entity soft matching algorithm to extract the ad-hoc path-formed explanations. However, these existing explainable product search models still have many shortcomings. First, the reasoning paths are typically explored based on the structure of the KGs and then converted into human-readable explanations using predefined templates. The exploration and conversion are not flexible enough to build correlations with diverse user queries, i.e., user search intent. Take an example in Figure 1(a), given the user's current query *universal high-quality headphones*, the reasoning path $\textit{User:ANS} \xrightarrow{query:\{electronic\ earphones\}} \textit{DJ Headphones} \xrightarrow{ProducedBy} \textit{Sony} \xrightarrow{ProducedBy^{-}}$ *Stereo Headphones* can give the explanation *the product is retrieved because the user has previously purchased products from brands such as Sony*. There is a weak semantic correlation between the path-formed explanation and the user's current query. Compared with the path-formed explanation, the natural language explanation *the product is a standard headphone that works on any device and is of high quality* can provide intuitive and easy-to-understand explanations for the user's current query through the keywords {*works on any device, high quality*}. This makes the natural language explanation more flexible to correlate with the user's search intent. Second, these models treat search and explanation as two independent tasks as shown in Figure 1(b), which leads to the gap between search results and explanation generation due to the insufficient interaction between them. Since the two tasks are highly coupled, they should be seamlessly integrated to share useful knowledge or features between them. Therefore, it is a challenge to generate human-readable explanations, such as review-like natural language sentences, within the context of fully exploiting and modeling search and explanation tasks.

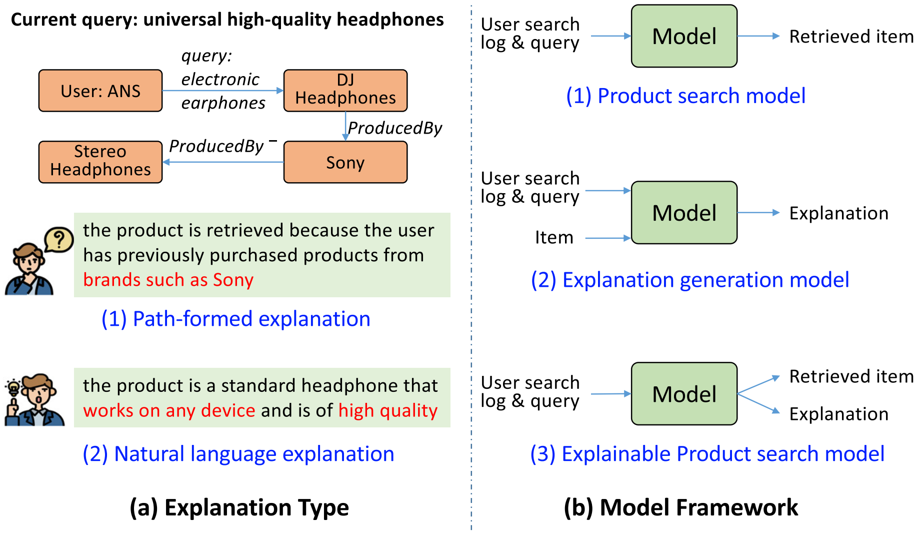
Current query: universal high-quality headphones
User: ANS
query: electronic earphones
DJ Headphones
ProducedBy
Stereo Headphones
ProducedBy⁻
Sony
the product is retrieved because the user has previously purchased products from brands such as Sony
(1) Path-formed explanation
the product is a standard headphone that works on any device and is of high quality
(2) Natural language explanation
(a) Explanation Type
User search log & query
Model
Retrieved item
(1) Product search model
User search log & query
Item
Model
Explanation
(2) Explanation generation model
User search log & query
Model
Retrieved item
Explanation
(3) Explainable Product search model
(b) Model Framework

Fig. 1. The illustration of different explanation types and model frameworks. The natural language explanation in (a) is more about semantic correlation for the current query than the path-formed explanation. The first two frameworks in (b) can only perform the search or generation task, while the latter can integrate both tasks into the same model and provide reasonable explanations for the retrieved products.

To address the above challenges, we aim to develop a more effective framework that integrates the search and explanation tasks in a unified manner, which can not only achieve accurate search performance, but also generate user-understandable explanations. Our model is inspired by the extensive application of **pre-trained language models (PLMs)** across various domains and a wide range of downstream tasks. For the limitation of path-based explanation, considering that PLMs have strong generation capabilities and can produce natural language sentences as explanations—a capability already validated by previous work [18, 50]—we regard PLMs as the backbone to perform the search task and generate natural language explanations simultaneously. For the gap between search results and explanation generation, inspired by the powerful adaptation capability to various downstream tasks of the PLMs, we leverage the PLMs as the backbone to integrate the search and explanation tasks in different settings, and use the search results to assist in explanation generation. Besides, the recently popularized paradigm of prompt learning [8, 11, 18] enables PLM to be applied to a variety of tasks in a low-cost and flexible manner. Prompt learning typically enhances and extends the input of PLMs using explicit or implicit prompt tokens in the form of instructions, learnable embeddings, or demonstrations, which can be concatenated to the original input or prefixed to the model's internal representation. By designing different prompts, this paradigm can unify various tasks or data using a single model. It only requires training the prompt parameters, thereby reducing the number of parameters needed to be trained and facilitating efficient training. Previous research [8, 51] has attempted to unify two different tasks using PLMs. For instance, in the field of conversational recommendation systems, efforts [52] have been made to utilize PLMs and prompts based on the same conversational semantics to achieve both recommendation and conversation objectives. In the areas of product search and explanation generation, both two subtasks leverage the same preference and query information to achieve different goals, making the development of a unified framework based on PLM and prompt learning feasible. Therefore, to harmonize the utilization of PLMs across diverse tasks in a straightforward yet adaptable way, we leverage the prompt learning technique [20, 27] to augment the task-specific input of PLMs, effectively catering to specific objectives.

To this end, we propose a unified prompt-aware framework for personalized product search and explanation generation, namely P-PEG, which designs the task-specific prompt generator to integrate task-specific knowledge into the base PLM more effectively. On the one hand, the unified framework can reduce the update of the PLMs' parameters for various downstream tasks, and effectively associate the two tasks seamlessly in a flexible manner. On the other hand, the task-specific prompt can offer adequate contextual information to adapt the PLM for the search and explanation generation tasks. To generate prompts that better meet the tasks, our prompt generator is to generate task-specific prompt inputs for the base PLMs by effectively utilizing user search history and the current query based on the adaptive fusion of the review context and knowledge graphs. The fusion can provide sufficient semantic information and structured knowledge about the background and context. In particular, for the search task, the generated search-specific prompts are used to assist the base PLM in extracting the user's search intents for a more accurate search. For the explanation task, the explanation-specific prompts together with the search results are the input of the base PLM, facilitating the generation of human-readable explanations. We develop the search results as part of the input for the explanation task, which can greatly enhance the information flow and interaction between the two tasks. It can also address the problem of semantic inconsistency between product search and explanation generation, which has not been well adequately tackled by previous methods. To use the prompt information more efficiently, we take the technique of P-tuning v2 [27] to incorporate the prompts into each layer of the base PLMs in the training stage. Moreover, we select GPT-2 [37] as our base PLM, remaining fixed throughout the training process without fine-tuning or continual pre-training. Empirical experiments on three Amazon datasets show that our P-PEG can generate user-friendly search results and reasonable natural language explanations.

In summary, our main contributions are as follows:

- We propose the P-PEG model that formulates the search and explanation tasks in the form of prompt learning, and designs specific prompts for each task in a unified manner.
- We leverage the search results as parts of the explanation-based prompts, which can boost the explanation generation and alleviate the gap between the two tasks.
- We design task-specific prompt generators to generate specific prompt inputs for the base PLMs through the adaptive fusion of text and knowledge information.

2 Related Work

2.1 Product Search

The goal of product search is to match queries with relevant items and provide a satisfying purchasing experience. Since the information about the products is often presented in a structured form, earlier models usually leveraged structured product facets such as categories and brands to extract item features of the products for retrieval [22, 46]. However, because the queries are often unstructured free-text, there is often a semantic gap between the queries and structured product features. To address this issue, researchers attempted to leverage the latent space model to learn the representations of products, queries, and users. For instance, Van Gysel [45] proposed a Latent Semantic Embedding model that extracts words from queries and product descriptions using n-gram and projects them into the same latent semantic space. HEM [3] constructed a hierarchical language model to model users and products using language models, and calculates the similarity between the items and the composition of query and user. Besides, researchers attempt sequence modeling to better meet users' personalized search interests and needs. For example, ZAM [1] adopted a novel attention mechanism to determine the weights of historical items for different user-query pairs and obtain the interests of users. Guo [13] introduced ALSTP which makes use of the attention mechanism and RNN to model the long- and short-term preferences of users, and combine them with the submitted queries as the user's intent. Because transformers have a strong ability to encode sequences and understand texts, TEM [5] used a transformer network to encode the query and the historical bought products, so that the products can be sorted according to the user's interests better. RTM [49] also proposed a transformer model that encodes the submitted query, user reviews, and item reviews in a unit to calculate the purchase score.

Although these methods are effective in dealing with unstructured text and sequence data, they are limited in structured metadata and relational information such as item co-occurrence relations or item co-occurrence relations. To address this limitation, DREM [4] incorporated knowledge graphs and metadata, such as categories into the model and leveraged the paths as explanations of the search results, which is the first attempt at the explainable product search. After that, DREM-HGN [2] extended the DREM model by using a hierarchical gated network to assign weight coefficients on the user's neighborhoods for constructing user representations. CAMI [26] proposed a category-aware multi-interest model, which uses entities in KG including brands and categories to explore users' multiple preferences. Although these models can effectively fit users' interests and improve search performance, they cannot provide a free-text explanation for search results that users care about.

2.2 Explanation Generation

The Natural language generation task is widely used in various fields like recommender systems [17, 50] and dialog systems [52], and has many applications such as explanation generation [18] and review generation [44]. There are several types of explanations in the explanation generation task, such as knowledge graph path, pre-defined templates, ranked text, and natural language explanations. In the search scenario, there has not been enough work on generating natural language explanations. Previous path-based explainable search methods [2, 4] selected paths from user to product as explanations, which may lack flexibility and not always be conducive to the user's understanding. To address these limitations, researchers have proposed to generate natural language explanations that have received enough attention in the field of recommendation [18, 50]. These models are mainly divided into two lines. The former is to tune the language model for generating explanations. Previous works mostly relied on RNN or unpre-trained language models. For example, NETE [17] leveraged an improved GRU model and review features to generate template-controlled sentences. PETER [50] designed three tasks to train a Transformer [47] model to generate the explanations. The latter is to fix the model based on prompt learning. For instance, Li et al. [18] proposed the PEPLER model which uses prompt learning to fuse user and item IDs into a PLM. Inspired by the natural language explanations in the recommendation, we use a PLM and attempt prompt learning to search for the products and generate the natural-language explanations tailored for user search intent. Currently, Large Language Models like ChatGPT [33] are quite popular and have strong capability of generation. It might be a potentially feasible approach for generating explanations.

2.3 Knowledge Embedding

Knowledge embedding is a commonly used method for modeling multi-relational data, primarily by building embeddings for entities and relations and modeling the associations between them. It is mainly aimed at learning the patterns among entities, which is used to model the knowledge of entities and relationships for subsequent tasks. Early methods for handling multi-relational data were primarily based on matrix factorization [14, 21, 32, 40]. These approaches involved factorizing tensors or decomposing user–item and item–item matrices to predict relations and model the knowledge embeddings of entities and relations. Some other researchers [31, 57] proposed to use non-parametric Bayesian frameworks for tasks such as link prediction, which also addressed the issues of multi-relational data modeling. Recently, more researchers [7, 41] tried to explore modeling multi-relational data using neural networks and knowledge graphs. Many studies [6, 24, 53] designed translation-based models to learn the knowledge of entities and predict the relations between entities. Furthermore, Ai et al. [4] and Liu et al. [26] leverage the translation-based methods to model the knowledge embeddings of users, queries, and products to predict the next product that a user might purchase. In our work, we utilize the knowledge embeddings of users, queries, and products to capture their structured and collaborative information. We combine this information with textual semantic information to construct prompts, thereby accomplishing the tasks of product search and explanation generation.

3 Problem Statement

Our model P-PEG aims to search for suitable products and generate explanations using the shared PLM. To better extract users' search intent and preference, P-PEG takes a user u, a submitted query q, and the user's behavior sequence $H = \{q_1, i_1, q_2, i_2, ., q_k, i_k\}$ as input, and performs the following task: (1) search task is to select the most appropriate product $\hat{i}$ that meets the user's search intent under the given query q, and (2) explanation task is to generate the reasonable natural-language explanation $\hat{E}$ for the product $\hat{i}$ tailed for user's search intent.

4 Method

In this section, we present a unified prompt-aware framework for personalized product search and explanation generation, namely P-PEG, which implements the search and explanation generation tasks in a unified manner. Our model mainly consists of three components: the base PLM, the personalized search component, and the explanation generation component. The base PLM is the foundation of the entire model that can accomplish the search and explanation tasks in a unified framework. The search and explanation components each include a specific prompt generator to assist with retrieving products or generating solid explanations based on the base PLM. Moreover, aiming to enhance the interaction between the two tasks, we develop the search semantic signal from the search component as part of the input for the explanation task to alleviate the gap between the search results and explanation generation. In this way, the search and explanation tasks can be fulfilled in a unified approach, which can make accurate search results and generate convincing explanations simultaneously. The overall framework of our model is shown in Figure 2.

4.1 The Base PLM

The paradigm of prompt learning utilizing PLM is to reformulate the original input of base PLMs by prefixing latent or explicit tokens in different settings, which has been witnessed remarkable performance on a variety of fields, such as recommendation [18, 52], machine translation [8, 42], classification [11, 12, 15], computer vision [30], etc. In the recommendation scenario that resembles our task, this paradigm has successfully been used to generate review-like explanations in explainable recommendation [18], where the generated explanations are reasonable and coherent for recommendation results. Therefore, we use $f(\cdot \mid \theta_{\text{PLM}}; \mathbf{G})$ to represent the base PLM parameterized by θ_{PLM}, which takes a token sequence G as input and creates contextualized representations for each token. In this article, to align with baselines [18], we choose GPT-2-base [36, 37] as our backbone PLM, which can generate coherent and contextually relevant text [28], making it useful for various NLP tasks such as summarization [54] and question answering [39]. Focusing on our task, G is the prompt learning based input to the PLM obtained from the search and explanation

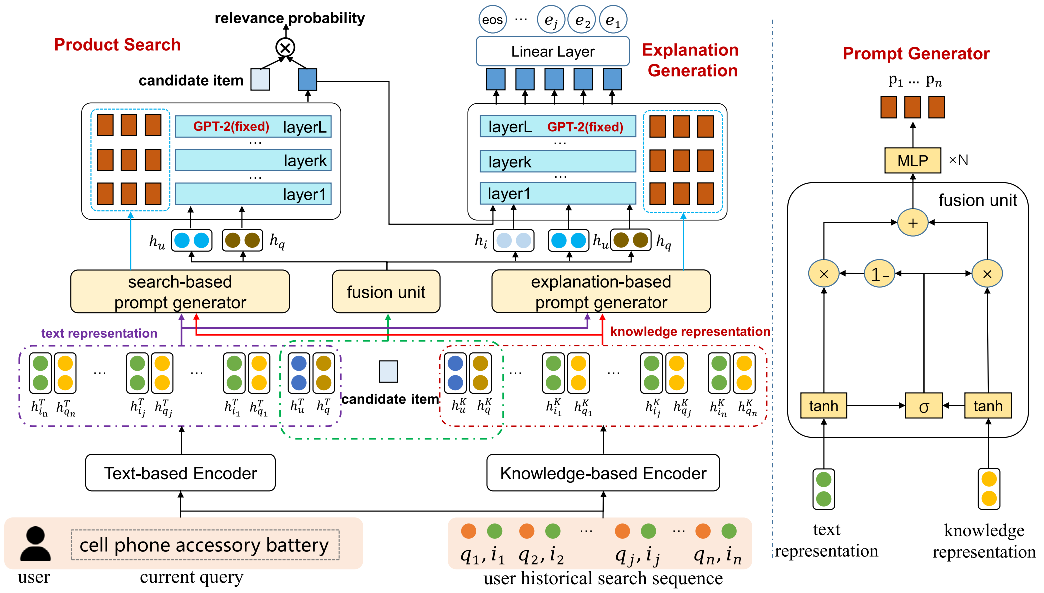

Fig. 2. The main architecture of our proposed model P-PEG on the left side of the dotted line. It consists of two components, including the search part (left) and the explanation generation part (right). These components share the text-based encoder and the knowledge-based encoder. First, we perform pre-training to obtain semantic representations of text and KGs in the text-based and knowledge-based encoders. Second, we fuse the text and knowledge semantics in the search-specific prompt generator, and then prompt the PLM to perform the search task. Finally, we use the search results as part of the explanation prompt for explanation generation.

component. Because GPT2 does not have a CLS token, we will use the contextualized representation of the last token to perform search and explanation generation tasks.

4.2 Personalized Search Component

The core of the personalized search component is to develop a search-based prompt generator to assist the base PLM in performing the search task. The search-specific prompt generator takes the user u, the current query q, and the user historical search sequence H as input, and generates l search-specific prompts as the input of the base PLM. To enrich the prompt representations, our prompt generator fuses the text-based and knowledge-based semantic information of the input sequence $\{u, q, H\}$ from the text-based encoder and knowledge-based encoder respectively. In this way, our search component takes these prompts as the input of the base PLM, and can retrieve the suitable products given the user's current query. In the following, we first introduce the text-based and knowledge-based representation encoder and then present the design of our prompt generator. After that, we introduce the method of prompt learning.

4.2.1 Representation Encoder. The representation encoder aims to obtain the high-level semantic representations of the user u, the current query q, and the search historical sequence H, which can be fed into the prompt generator for generating high-quality prompts. First, we consider using textual information in the model because the texts associated with users and products, including queries, reviews, titles, and categories, contain a wealth of key information. This information plays a crucial role in both the search and explanation generation process, which has been verified by previous works [3, 10]. Therefore, for the textual information, we directly use BERT [9] as a text-based encoder to encode the text like user reviews and product's title/description of the input sequence $\{u, q, H\}$ for getting the text-based semantic representations. Thus, the text-based encoder can be defined as:

$$\mathbf{h}_u^{\text{text}}, \mathbf{h}_q^{\text{text}}, \mathbf{h}_i^{\text{text}} = \text{TextEncoder}\left(T(u), T(q), T(i)\right), \tag{1}$$

where TextEncoder$(\cdot)$ is BERT [9], $T(\cdot)$ represents the text sequence of the input, i.e., $T(u)$ is the user review texts on his purchased products, $T(i)$ is the review texts that the product i has received from users, and $T(q) = [w_1; w_2; ...; w_{n_q}]$ is the words in the query, $[;]$ is the concatenation operation, and $\mathbf{h}_u^{\text{text}}$, $\mathbf{h}_q^{\text{text}}$, $\mathbf{h}_i^{\text{text}} \in \mathbb{R}^{d_1}$ are [CLS] representation of the encoder for the user u, the query q and the item i.

Second, we consider incorporating information from the knowledge graph into the model because the KG contains a wealth of structured and collaborative information. By encoding the entities and relations in the KG, we can obtain their structured knowledge, which has been applied in most of the promising models in product search [4, 26]. For the knowledge-based encoder, we consider the structural properties of user–product KG and extract the high-order connectivity information of the user, the query, and the products as knowledge-level representations. We utilize the promising KG-based product search method DREM [4] as our knowledge-based encoder, and define it as:

$$\mathbf{h}_u^{\text{KG}}, \mathbf{h}_q^{\text{KG}}, \mathbf{h}_i^{\text{KG}} = \text{KGEncoder}(u, q, i), \tag{2}$$

where KGEncoder$(\cdot)$ is the DREM model [4] that expects the high plausibility of the triplet (u, q, i), $\mathbf{h}_u^{\text{KG}}$, $\mathbf{h}_q^{\text{KG}}$, $\mathbf{h}_i^{\text{KG}} \in \mathbb{R}^{d_2}$ are the knowledge embeddings of the user u, the query q and the product i.

Importantly, the *query* is regarded as a dynamic relation between users and products in DREM because its semantic is determined by the search content $\{w_1, w_2, ..., w_{n_q}\}$. Thus, we use the weighted sum of word embedding and a non-linear projection function similar to DREM to obtain the query embedding as:

$$h_q^{\text{KG}} = \tanh\left(\mathbf{W} \cdot \frac{\sum_{w_q \in q} \mathbf{w}_q}{n_q} + \mathbf{b}\right), \tag{3}$$

where n_q is the length of the query, $\mathbf{w}_q \in \mathbb{R}^{d_2}$ is word embedding in the query learned by DREM, $\mathbf{W} \in \mathbb{R}^{d_2 \times d_2}$ and $\mathbf{b} \in \mathbb{R}^{d_2}$ are the parameters learned by DREM to obtain the query embeddings. In addition, our text-based and knowledge-based encoders are pre-trained and not updated during training to reduce the scale of parameters and run the model more efficiently.

4.2.2 Search-Specific Prompt Generator. Based on the fixed base PLM, the search-specific prompt generator aims to generate a sequence of continuous task-specific vectors and prepends it to the input of the base PLM, which can adapt the base PLM to the search tasks. Consequently, we only need to store one copy of the large PLM and learn the small task-specific prompts for optimizing the search task. In the design of the prompts, to combine the advantages of the two representations and complement them, our search-specific prompt generator fuses the text-based and knowledge-based representations of input sequence $\{u, q, H\}$ to clarify user preferences. Specifically, the prompt generator consists of two steps: (1) the gate-based fusion on the text-based and entity-based representations, which learns a cross-interaction network to associate the two kinds of representations through the adaptive gate network. (2) Transformation on the fused representations into task-specific prompts, which learns a multi-layer **multilayer perceptron (MLP)** transformation to obtain the prompts. For the input sequence $\{u, q, H = \{q_1, i_1, ..., q_k, i_k\}\}$, its text-based and knowledge-based representations from the text-based and knowledge-based encoder are denoted as:

$$\begin{aligned}\mathbf{S}^{\text{text}} &= \left[\mathbf{h}_u^{\text{text}}; \mathbf{h}_q^{\text{text}}; \mathbf{h}_{q_1}^{\text{text}}; \mathbf{h}_{i_1}^{\text{text}}; ...;\ \mathbf{h}_{q_k}^{\text{text}}; \mathbf{h}_{i_k}^{\text{text}}\right], \\ \mathbf{S}^{\text{KG}} &= \left[\mathbf{h}_u^{\text{KG}}; \mathbf{h}_q^{\text{KG}}; \mathbf{h}_{q_1}^{\text{KG}}; \mathbf{h}_{i_1}^{\text{KG}}; ...; \mathbf{h}_{q_k}^{\text{KG}}; \mathbf{h}_{i_k}^{\text{KG}}\right],\end{aligned} \tag{4}$$

where $[;]$ is the concatenation operation, $\mathbf{S}^{\text{text}} \in \mathbb{R}^{(2k+2)\times d_1}$, $\mathbf{S}^{\text{KG}} \in \mathbb{R}^{(2k+2)\times d_2}$. Then the cross-interaction mechanism on the gated-based fusion function *Fusion*$(\cdot)$ is as follows:

$$\begin{aligned}\mathbf{h}^{\text{KG}} &= \tanh(\mathbf{W}_k \mathbf{S}^{\text{KG}}), \\ \mathbf{h}^{\text{text}} &= \tanh(\mathbf{W}_t \mathbf{S}^{\text{text}}), \\ \alpha &= \sigma(\mathbf{W}_f [\mathbf{h}^{\text{KG}}, \mathbf{h}^{\text{ text}}]), \\ \mathbf{h}^{\text{fuse}} &= \alpha \odot \mathbf{h}^{\text{KG}} + (1-\alpha) \odot \mathbf{h}^{\text{text}},\end{aligned} \tag{5}$$

where $\mathbf{W}_k \in \mathbb{R}^{d_2 \times d_1}, \mathbf{W}_t \in \mathbb{R}^{d_1 \times d_1}, \mathbf{W}_f \in \mathbb{R}^{2d_1 \times d_1}$ are model parameters to be learned. From Equation (5), we can see that α is an adaptive parameter that controls the proportion of knowledge representation versus text representation in the fused representation. When α is small, the output of the fusion unit is dominated by the text representation, whereas when α is large, the knowledge-based representation will take a large proportion.

After the fusion step, we perform a *Generator*$(\cdot)$ function to transform the fusion representation into l continuous prompts. Here we use MLP as the *Generator*$(\cdot)$, i.e.:

$$\mathbf{P}_{\text{sch}} = \text{ReLU}\left(\mathbf{W}_n(...\text{ReLU}(\mathbf{W}_1 \cdot \mathbf{h}^{\text{fuse}} + \mathbf{b}_1)...) +\ \mathbf{b}_n\right), \tag{6}$$

where $\{\mathbf{W}_1, ..., \mathbf{W}_n\}$ and $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$ are learnable parameters. The result can be represented as $\mathbf{P}_{\text{sch}} = [\mathbf{p}_{\text{sch}}^1, \mathbf{p}_{\text{sch}}^2, ..., \mathbf{p}_{\text{sch}}^l]$, which are the search-based prompts used in the search task subsequently.

4.2.3 Prompt Learning. Since the search task aims to retrieve some suitable products for the user under a given query. To acquire the rich information of the user and query, we fuse the text-based and knowledge-based representation of them using Equation (5), and denote them as $\mathbf{h}_u^{\text{fuse}}$ and $\mathbf{h}_q^{\text{fuse}}$. In general, we take them and the search-based prompts $\mathbf{P}_{\text{sch}}$ as input for the prompt learning:

$$\mathbf{G}_{\text{sch}} = \left[[\mathbf{h}_u^{\text{fuse}};\ \mathbf{h}_q^{\text{fuse}}],\ \mathbf{P}_{\text{sch}}\right], \tag{7}$$

where $\mathbf{h}_u^{\text{fuse}}$ and $\mathbf{h}_q^{\text{fuse}}$ are direct inputs for the base PLM, and the search-based prompts $\mathbf{P}_{\text{sch}}$ are added as prefix tokens to each layer of GPT-2 in the same way as [27], which allows for better tuning and allows the prompts to have a deeper and more direct impact on the model's predictions.

In the above search task component, only the generated prompts $\mathbf{P}_{\text{sch}}$, and the fused input representation $\mathbf{h}_u^{\text{fuse}}, \mathbf{h}_q^{\text{fuse}}$ are to be learned. We denote them as θ_{sch}. For learning θ_{sch}, we utilize the widely used cross-entropy loss as our objective function:

$$\mathcal{L}_{\text{sch}} = -\sum_{i=1}^{N}\sum_{j=1}^{M}\left(y_{i,j} \cdot \log P(j) + \left(1 - y_{i,j}\right) \cdot (1 - \log P(j))\right), \tag{8}$$

where N is the number of training samples and M is the number of products, $y_{i,j}$ is a binary label that equals 1 when the item j is the right label given user and query, and $P(j) = P(j|u, q; \theta_{\text{sch}})$ is the predicted scores through softmax function.

4.3 Explanation Generation Component

The explanation generation component mainly gives the user an acceptable natural language explanation according to the user's relevant information and the query content. The architecture of the explanation generation component is mainly composed of an explanation-based prompt

Algorithm 1: The Training Process of P-PEG

```
1:  Input: the user $u$, the current query $q$, user's behavior sequence $H = \{q_1, i_1, q_2, i_2, ..., q_k, i_k\}$, the item set $I$, the search component $\pi_S$, and the explanation generation component $\pi_E$.
2:  Output: the retrieved product $\hat{i}$ and the generated explanations $\hat{E}$ for the product $\hat{i}$.
    Search component
3:  for k = 1, ... epoch do
4:      Initialize the total loss $L_{sch} = 0$.
5:      \\ Text representation
6:      $\mathbf{h}_u^{\text{text}} = \text{TextEncoder}(T(u))$
7:      $\mathbf{h}_q^{\text{text}} = \text{TextEncoder}(T(q))$
8:      $\mathbf{h}_{q_j}^{\text{text}} = \text{TextEncoder}(T(q_j)), j \in [1, k]$
9:      $\mathbf{h}_{i_j}^{\text{text}} = \text{TextEncoder}(T(i_j)), j \in [1, k]$
10:     $\mathbf{S}^{\text{text}} = [\mathbf{h}_u^{\text{text}}; \mathbf{h}_q^{\text{text}}; \mathbf{h}_{q_j}^{\text{text}}; \mathbf{h}_{i_j}^{\text{text}}], j \in [1, k]$
11:     \\ Knowledge representation
12:     $\mathbf{h}_u^{\text{KG}} = \text{KGEncoder}(u)$
13:     $\mathbf{h}_q^{\text{KG}} = \text{KGEncoder}(q)$
14:     $\mathbf{h}_{q_j}^{\text{KG}} = \text{KGEncoder}(q_j), j \in [1, k]$
15:     $\mathbf{h}_{i_j}^{\text{KG}} = \text{KGEncoder}(i_j), j \in [1, k]$
16:     $\mathbf{S}^{\text{KG}} = [\mathbf{h}_u^{\text{KG}}; \mathbf{h}_q^{\text{KG}}; \mathbf{h}_{q_j}^{\text{KG}}; \mathbf{h}_{i_j}^{\text{KG}}], j \in [1, k]$
17:     \\ Fusion representation
18:     $\mathbf{h}^{\text{fuse}} = \text{Fusion}(\mathbf{S}^{\text{text}}, \mathbf{S}^{\text{KG}})$
19:     $\mathbf{h}_u^{\text{fuse}} = \text{Fusion}(\mathbf{h}_u^{\text{text}}, \mathbf{h}_u^{\text{KG}})$
20:     $\mathbf{h}_q^{\text{fuse}} = \text{Fusion}(\mathbf{h}_q^{\text{text}}, \mathbf{h}_q^{\text{KG}})$
21:     \\ Search-specific prompts
        $\mathbf{P}_{\text{sch}} = \text{Generator}(\mathbf{h}^{\text{fuse}})$
22:     \\ Prompt learning for search task
23:     retrieved item $\hat{i} = \text{argmax}_{i \in I}(\text{GPT}([\mathbf{h}_u^{\text{fuse}}; \mathbf{h}_q^{\text{fuse}}], \mathbf{P}_{\text{sch}})$
24:     minimize the loss $L_{sch}$ to update $\pi_S$.
25: end for
    Explanation component
26: for k = 1, ... epoch do
27:     Initialize the total loss $L_{exp} = 0$.
28:     perform line 5-20
29:     \\ Explanation-specific prompts
30:     $\mathbf{P}_{\text{exp}} = \text{Generator}(\mathbf{h}^{\text{fuse}})$
31:     search intentions $\mathbf{h}_s = \text{GPT}([\mathbf{h}_u^{\text{fuse}}; \mathbf{h}_q^{\text{fuse}}], \mathbf{P}_{\text{sch}})$
32:     generate $\hat{E} = \text{GPT}([\mathbf{h}_u^{\text{fuse}}; \mathbf{h}_q^{\text{fuse}}; \mathbf{h}_s], \mathbf{P}_{\text{exp}})$.
33:     minimize the loss $L_{exp}$ to update $\pi_E$.
34: end for
35: Return $\pi_S$, $\pi_E$, $\hat{i}$, $\hat{E}$.
```

generator and the semantic signal from the search component. Here, the explanation-based prompt generator is similar to the search-based prompt generator. The following gives the details of our explanation generation component.

The input of the explanation component consists of user u, query q, label item i, behavior sequence $H = \{q_1, i_1, ., q_k, i_k\}$. We use the same encoders as the search component in Equations (1) and (2) to encode them and get the text-based and knowledge-based representation of the input. We take them as the input of the explanation-based prompt generator and use Equation (5) to merge and transform the representations into explanation-based prompts:

$$\mathbf{P}_{\text{exp}} = \text{ReLU}\left(\mathbf{W}'_{n'}(\ldots\text{ReLU}(\mathbf{W}'_1 \cdot \mathbf{h}^{\text{fuse}} + \mathbf{b}'_1)\ldots) + \mathbf{b}'_{n'}\right), \tag{9}$$

where $\{\mathbf{W}'_1, ..., \mathbf{W}'_{n'}\}$ and $\{\mathbf{b}'_1, ..., \mathbf{b}'_{n'}\}$ are the parameters to be learned, $\mathbf{P}_{\text{exp}} = [\mathbf{p}^1_{\text{exp}}, \mathbf{p}^2_{\text{exp}}, ..., \mathbf{p}^{l'}_{\text{exp}}]$ is the explanation prompts with the number of l'. The explanation generator has the same architecture as the search-based prompt generator but has different parameters.

Since the explanation task aims at generating an explanation to the user for the retrieved products under the submitted query, we take the fused representation of the user $\mathbf{h}^{\text{fuse}}_u$, query $\mathbf{h}^{\text{fuse}}_q$ and item $\mathbf{h}^{\text{fuse}}_i$ as part of the input using the same encoder and fusion unit. Moreover, to bridge the gap between the search and explanation task and leverage the information of the search task effectively, we use the last hidden state vector of the search component $\mathbf{h}_s$, which represents the user's search intentions and preferences, as part of the input for the GPT-2 in the explanation component:

$$\mathbf{G}_{\text{exp}} = \left[[\mathbf{h}^{\text{fuse}}_u;\ \mathbf{h}^{\text{fuse}}_q;\ \mathbf{h}^{\text{fuse}}_i;\ \mathbf{h}_s], \mathbf{P}_{\text{exp}}\right], \tag{10}$$

where $\mathbf{h}^{\text{fuse}}_u$, $\mathbf{h}^{\text{fuse}}_q$, $\mathbf{h}^{\text{fuse}}_i$, and $\mathbf{h}_s$ are the direct inputs for the base PLM, the use of explanation-based prompts $\mathbf{P}_{\text{exp}}$ is similar to that in the search component.

In the explanation generation component, it is the auto-regressive generation objective to generate the review-like explanation sentences for the retrieved items. Specifically, the output vectors from the last layer of GPT-2 are processed through a linear layer to create a probability distribution encompassing all tokens within the dataset. This distribution is then utilized to predict the next token based on the preceding ones. It is achieved by minimizing the negative log-likelihood:

$$\begin{aligned}\mathcal{L}_{\text{exp}} &= -\frac{1}{N}\sum_{i=1}^{N}\log P(E_i \mid \theta_{\text{exp}}) \\ &= -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{L_i}\log P(w_{ij} \mid \theta_{\text{exp}}; w_{<j}),\end{aligned} \tag{11}$$

where $\theta_{\text{exp}} = \{\mathbf{P}_{\text{exp}}, \mathbf{h}^{\text{fuse}}_u, \mathbf{h}^{\text{fuse}}_q, \mathbf{h}^{\text{fuse}}_i\}$ are the learnable variants. w_{ij} is the next token to be predicted for the training sample i, $w_{<j}$ denotes the word before the jth position, and L_i is the length of the explanation for sample i.

4.4 Model Optimization

The process of our model P-PEG is described as Algorithm 1. It takes the given user u, the current query q and user's behavior sequence $H = \{q_1, i_1, q_2, i_2, ., q_k, i_k\}$ as input. As output, it delivers the optimized search component π_S, explanation generation component π_E, the most appropriate product $\hat{i}$ and the generated explanations $\hat{E}$ for the product $\hat{i}$. Specifically, the parameters of P-PEG are composed of three groups including two task-specific prompt generators and the base PLM, which are represented as θ_{sch}, θ_{exp}, and θ_{PLM} respectively. During our training process, the parameters of our PLM θ_{PLM} are fixed, and we train the other two tasks in a pipeline, including: (a) we first encode the input information of users, queries, items, and history behavior sequences using a pre-trained text-based encoder and knowledge-based encoder to obtain text-based and knowledge-based representations, (b) we fuse text-based and knowledge-based representations of user and query and further generate search prompts using Equations (5) and (6). Later we input them to the base PLM to train the parameter θ_{sch} using the Equation (8) for the search task, and (c) lastly, we generate the explanation prompts based on the fusion of the representations of user, query, and item using Equations (5) and (6), and input them together with search semantic signal vector $\mathbf{h}_s$ from the search component to the base PLM to train parameter θ_{exp} using Equation (11).

Table 1. Statistics of the Three 5-Core Datasets

	Cellphones and Accessories	Electronics	CD and Vinyl
#Users	27,879	192,403	75,258
#Products	10,429	63,001	64,443
#Queries	165	989	674
#Records	194,439	1,689,188	1,097,591
#Words/Explanation	13.51	12.98	12.76
#(u,q) pairs	113,855	1,212,077	1,342,775
#(u,q,i) samples	173,681	1,351,011	3,387,278
#Relevant items per pair	1.53	1.11	2.52
#density	1.46E-04	1.77E-05	3.91E-05
#average review length	93.50	118.27	174.57

5 Experimental Setup

5.1 Dataset

We choose the Amazon dataset[1] as our experiment dataset, which is a popular dataset and widely used in the product search field [2, 3, 26] and recommendation field [17, 18]. It contains the information of users, products, their purchase relations, and other metadata such as users' reviews, products' brands, categories, titles, and descriptions. To make the experiments more reliable and to align with the previous work [4, 26], we use 5-core data, in which each user and each item has at least 5 associated reviews. We choose three subsets of Amazon for our experiments, including Electronics, Kindle Store, CDs and Vinyl, and Cell Phones and Accessories. Following [4, 26], query words are extracted from the categories of the product and then concatenated into sentences as queries. Statistics of the datasets are shown in Table 1.

For the choice of labels of explanation sentences, previous works [17, 50] have shown that sentences in reviews can be used as an explanation of high quality, so we adopted a similar approach to them. We leverage an effective toolkit [56], which can extract (feature, opinion, sentiment) triplets from free-text corpus such as reviews, to get the key feature words from users' reviews. In this process, from the review of each purchase behavior, we can get the feature and sentiment words that contain the user's attitude towards the product, and cut off the sentences that contain these words and use them as an explanation label for the purchase.

5.2 Evaluation Metrics

Following previous works of product search field [4, 26], we adopt three metrics including **mean average precision (MAP)**, **mean reciprocal rank (MRR)**, and **normalized discounted cumulative gain at 10 (NDCG@10)** to evaluate the retrieval performance of our search model. For the explanation generation task, to better measure the accuracy of the generated explanations, we evaluate the generated explanations from two aspects, including text quality and explainability. For the text quality, we choose BLEU [34] and BERTScore [55] in machine translation and ROUGE [23] in text summarization, and report BLEU-1 and BLEU-4, and Precision, Recall, and F1-score of ROUGE-1, ROUGE-2 and BERTScore.

Moreover, because the above metrics only focus on gram-level precision and recall, they can not fully evaluate the generated sentences. To better measure the explanation sentences, following [17, 18], we adopt four other evaluation metrics:

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

(1) **Unique sentence ratio (USR)** computes the ratio of unique generated explanation sentences:

$$\mathrm{USR} = \frac{|S|}{N}, \tag{12}$$

where S denotes the set of unique generated sentences, and N represents the total number of test sentences.

(2) **Feature matching ratio (FMR)** measures the matching ratio between the words in the generated sentences and the label texts:

$$\mathrm{FMR} = \frac{1}{N} \sum_{\tau} \delta(f_\tau \in \hat{S}_\tau), \tag{13}$$

where $\tau = (u, q, i)$, f_τ is the features extracted from the ground-truth text, $\hat{S}_\tau$ is the explanation generated by the model for a given user-query-item pair, and δ is an indicator function: $\delta(x) = 1$ if x is true, and $\delta(x) = 0$ otherwise.

(3) **Feature coverage ratio (FCR)** measures how much the features from the generated sentences in the features of the ground truth:

$$\mathrm{FCR} = \frac{N_g}{|\mathcal{F}|}, \tag{14}$$

where N_g is the number of unique features in the generated sentences, and $\mathcal{F}$ is the set of distinct features in the label explanation sets.

(4) **Feature diversity (DIV)** measures the feature diversity of all generated explanations. For different (u, q, i) triplets, we hope that the explanation can cover more features. So it computes the intersection of feature sets between any two generated explanations:

$$\mathrm{DIV} = \frac{2}{N \times (N-1)} \sum_{\tau, \tau'} \left| \hat{\mathcal{F}}_\tau \cap \hat{\mathcal{F}}_{\tau'} \right|, \tag{15}$$

where $\tau = (u, q, i)$, $\tau' = (u', q', i')$, $\hat{\mathcal{F}}_\tau$ and $\hat{\mathcal{F}}_{\tau'}$ denotes the feature sets of the two generated explanations for different test samples, and $|\cdot|$ computes the number of features in the set. Among these metrics, DIV will be better if it is lower. The other metrics will be better when they are higher.

5.3 Baselines

We consider comparing the search and explanation generation results separately and choosing baselines for the two subtasks. For the product search task, we introduce six state-of-the-art baselines:

BM25. The classic probabilistic retrieval model BM25 [38] uses a 2-Poisson distribution to incorporate within-query term frequency, within-document term frequency, and document length to construct a score function for information retrieval.

LSE. LSE [45] is a latent space model for product search. It leverages a generative model to learn the representation of words and items. It computes the similarity scores between the queries and products to rank the items.

HEM. The personalized product search model HEM [3] adopts a hierarchical embedding model and uses the information of reviews to learn the semantic embeddings of users, products, and queries. It uses a language model to learn the representation of users and items and maximizes the likelihood of the generated reviews and ranks according to the probability of purchasing items.

DREM. DREM [4] is an explainable personalized product search model. Based on the knowledge graph, it constructs a dynamic purchasing relationship “search and purchase” between users and products, and ranks the products that meet the user’s query based on the relationship.

DREM-HGN. DREM-HGN [2] is an explainable personalized product search model, which extends DREM with a Hierarchical Gated Network. It generates model-intrinsic explanations for product search results using the attention weights extracted from HGN.

CAMI. CAMI [26] is a state-of-art retrieval model for personalized product search. It splits the user's embedding vector into several sub-vectors related to the category, which are used to express the user's multiple interests. The authors also propose a category-attention mechanism that combines multiple user representations of interests with queries and item representations to calculate a score.

For the explanation generation task, since there is no baseline for explanation generation in the product search field, we introduce several baselines from the product recommendation and partially modify them to fit the product search scenario.

Transformer. Transformer [47] is a classic sequence-to-sequence model consisting of encoders and decoders. We input the user and item ID as a word with the query into the transformer model, resulting in an appropriate explanation for the item.

NRT. NRT [19] is a Neural Rating and Tips generation model for recommendation. It uses user and item ID as input and applies a multi-layer perceptron to predict the rating and adopts a GRU model to generate a tip. In our task, we replace the output tip here with our explanation. As reported in [17, 50], there is a problem with generating identical sentences in the model, which is caused by the L2 regularization, and we remove it to be fair.

Att2Seq. Att2Seq [10] is a product review generation model using the given attributes. It uses a two-layer LSTM as a decoder to generate reviews, and we replace the generated review with our target explanation in our task. Following [17, 18], we remove the attention module as this module causes the generated content unreadable.

PETER. PETER [50] is a personalized recommendation and explanation generation model using an unpretrained transformer. It utilizes user and item IDs to predict ratings and generate an explanation, and also designs a context prediction task to bridge the gap between ID and words and avoid identical sentences.

PEPLER. PEPLER [18] is an explanation generation model utilizing the prompt learning approaches in the PLM. It fixes the PLM and designs two kinds of prompts including discrete prompts and continuous prompts. To be fair, we use the continuous prompt model as our baseline, which defines two trainable embedding matrices for user and item and doesn't include the feature words in the input.

For all baselines for the explanation task, to better fit the search scenario, we encode the query terms as part of the input to generate the explanations based on their own built-in vocabulary and word encoders.

5.4 Implementation Details

For the backbone PLM, we use GPT-2-base [37] to complete our experiments. For the data split, we follow [4, 26] and divide the data into the training set (70%) and the testing set (30%). We use the data split on all of the baselines for the search task and explanation task. We rerun HEM, DREM, CAMI, NRT, Att2Seq, PETER, and PEPLER using their released codes, and implement the rest baselines. For the baselines of the explanation task, we set the vocabulary size as 20,000 for Cellphones and Electronics, and 50,000 for CD and Vinyl. We also run five times on three datasets for baselines and report the average explanation performance in our experiment. We set the length of explanations to 20.

For our P-PEG, we select GPT-2 as our base PLM and freeze its parameters during training, which aims to make a fair comparison to the explanation baselines like PEPLER. We set the hidden size of prompts as 768 to be consistent with GPT-2, the size of text-based representation $d_1 = 768$, and

Table 2. The Hyper-parameter Settings of Product Search Baselines

Model	bs	Learning Rate	Embed Size	Clip Grad Norm	Neg Num	L2 Regularization	Dynamic Relation Weight
HEM	64	0.5⟶0	100 to 500	5	5	0.005	-
DREM	64	0.5⟶0	100 to 500	5	5	0.005	0.1 to 0.5
DREM-HGN	64	0.5⟶0	100 to 500	5	5	0.005	0.1 to 0.5
CAMI	64	0.5⟶0	200	5	5	0.005	0.1 to 0.5

Table 3. The Hyper-parameter Settings of Explanation Generation Baselines

Model	bs	Learning Rate	Embed Size	Hidden Unit Num	Length of E	Token Num	Dim of FFN	Clip Grad
Transformer	128	1⟶0.25	512	-	20	50,257	2,048	1
NRT	128	0.001	300	400	20	50,257		-
Att2Seq	100	0.001	64	512	20	50,257	-	5
PETER	128	1⟶0.25	512	-	20	50,257	2,048	1
PEPLER	128	0.001	768	-	20	50,257	-	-

the size of knowledge-based representation $d_2 = 200$. Besides, we set the length of search-based prompts $l = 10$ and explanation-based prompts $l' = 20$, the number of layers of MLP in prompt generator $n, n' = \{12, 345\}$ for different datasets. We utilize AdamW [29] to optimize the trainable parameters in P-PEG with the learning rate 0.0003 for the search task and 0.0001 for the explanation generation task.

For the hyper-parameter settings of baselines in the search task and the explanation generation task, we summarize them in Tables 2 and 3. The hyper-parameters of the product search baselines in Table 2 include **batch size (bs)**, learning rate, embedding size, clip gradient norm, **negative sample number (neg num)**, L2 regularization strength, and dynamic relation weight. Besides, the weight of query λ in HEM is tuned from 0.0 to 1.0. The hyper-parameter of temperature τ in CAMI is $\tau_{\max} = 3.0$ and $\tau_{\min} = 0.05$, and the size of multiple interest in CAMI is 4. The settings of BM25 scoring parameters k_1 and b are from 0.5 to 4 and 0.25 to 1. The hyper-parameters of explanation generation in Table 3 include bs, learning rate, embedding size, the number of hidden units, length of explanations (length of E), token number in the vocabulary list (token num), dimension of FFN and gradient clipping threshold (clip grad). Additionally, the number of LSTM layers in Att2Seq is 2, and the number of hidden layers in PEPLER is 2.

6 Experimental Results

In this section, we first present the experimental results and analysis on the search task and explanation generation task, and then introduce the ablation experiments, case studies, effectiveness assessment on the hyper-parameters, and user study.

6.1 Results Analysis on Search Task

The results of the product search task on three datasets are shown in Table 4. From the results we can find: (1) Our model outperforms the existing personalized and non-personalized models, where the relative performance improvement on NDCG@10 over them is at least 13.4%, 4.1%, and 14.7% on the three datasets respectively. This is mainly because our search-based prompt generator can capture accurate users' search interests by fusing text-based and knowledge-based representations. (2) Our model achieves a better performance than the graph-based methods (DREM, DREM-HGN, CAMI) on the Electronics dataset and achieves a comparable performance on the other two datasets. It indicates that our adaptive semantic incorporation of text representation can effectively model

Table 4. The Results on the Search Task of All Models on the Datasets

Model	Cellphones and Accessories			Electronics			CD and Vinyl		
	MAP@10	MRR@10	NDCG@10	MAP@10	MRR@10	NDCG@10	MAP@10	MRR@10	NDCG@10
BM25	0.083	0.081	0.115	0.283	0.280	0.304	0.027	0.018	0.016
LSE	0.098	0.098	0.084	0.233	0.234	0.239	0.018	0.022	0.020
HEM	0.124	0.124	0.153	0.308	0.309	0.329	0.034	0.040	0.040
DREM	0.249	0.249	0.282	0.367	0.367	0.392	0.067	0.077	0.078
DREM-HGN	0.252	0.252	0.288	0.369	0.369	0.401	0.073	0.081	0.086
CAMI	0.258	0.258	0.297	0.375	0.376	0.415	0.080	**0.089**	0.095
P-PEG (ours)	**0.263**	**0.263**	**0.337****	**0.386****	**0.386****	**0.432****	**0.089****	**0.089**	**0.109****

The best results are shown in bold. * and ** indicate the model outperforms all baselines significantly with paired *t*-test respectively for $p < 0.05$ and $p < 0.01$.

the user's preference and retrieve suitable products for users in some scenarios. (3) The personalized methods outperform non-personalized methods with remarkable improvement. It is because the former can effectively use the user's behavior sequence information to extract user personalized interests for ranking the products. (4) The graph-based methods have a higher search performance compared with non-graph-based methods. It suggests that the KG's structural information on the entities and relations is helpful in capturing users' interests for more accurate search performance.

6.2 Results Analysis on Explanation Task

The results of the explanation generation task on three datasets are shown in Table 5. To better evaluate the text quality of the explanation, we also use BERTScore to measure it and the results are shown in Table 6. In terms of text quality metrics BLEU, ROUGE, and BERTScore, (1) Our P-PEG significantly outperforms the five baselines on the three datasets because we leverage the search information to promote the explanation generation, which makes the generated sentences fit the search intention better. (2) Our model achieves better performance compared with Transformer-based and RNN-based methods (Transformer, PETER, NRT, Att2Seq). On one hand, the GPT-2 has a stronger generative ability than RNN and Transformer, which is suitable for our generation task and can generate more accurate and consistent sentences. On the other hand, our prompt generator provides more valuable information on each layer of GPT-2, helping the model generate better explanations with strong prompts. (3) Our model outperforms the prompt-based method PEPLER because we leverage the search information to bridge the gap between two tasks and promote information interaction. In addition, the incorporation of the knowledge-based and text semantic representations in the prompt generator makes our generated explanations closer to the user's search intent.

In terms of explainability metrics USR, FMR, FCR, and DIV, (1) Our model achieves the best results for almost all of the metrics on the three datasets. It is because we leverage the explanation-based prompts and search information to assist the GPT-2 to generate diverse and accurate sentences. Obviously, Transformer and NRT get a low value of USR, which infers that these methods often generate exactly the same explanation sentences in different samples. (2) Compared with RNN-based models, our P-PEG achieves better results because we leverage the prompt information that fuses the text-based and knowledge-based representations to assist the generation of GPT-2 effectively. Furthermore, NRT and Att2Seq have slightly inadequate sequence modeling capabilities due to the long-term dependency problem of RNN, resulting in bad performance on the explainability metrics. (3) PETER and PEPLER are better than RNN-based models in these metrics because the transformer can make the future tokens pay attention to the past tokens. Compared with them, our

Table 5. The Overall Performance of the Explanation Generation Task on the Three Datasets

Model	Text Quality								Explainability			
	B1↑	B4↑	R1-P↑	R1-R↑	R1-F↑	R2-P↑	R2-R↑	R2-F↑	USR↑	FMR↑	FCR↑	DIV↓
Cellphones and Accessories												
Transformer	10.27	0.42	12.88	11.41	11.33	1.34	1.11	1.12	0.18	0.07	0.02	0.63
NRT	10.15	0.28	14.65	10.62	11.45	1.37	0.94	1.02	0.08	0.05	0.01	0.46
Att2Seq	10.52	0.19	14.34	11.01	11.59	1.38	1.09	1.12	0.31	0.06	0.02	0.38
PETER	9.81	0.42	13.92	10.28	10.78	1.52	1.03	1.10	0.31	0.08	0.04	0.37
PEPLER	10.68	0.46	12.26	10.92	10.87	1.14	1.03	1.10	0.56	0.08	0.05	**0.19**
P-PEG (ours)	**11.07****	**0.72****	**15.25***	**11.97****	**12.21****	**1.82***	**1.44***	**1.41***	**0.89****	**0.12****	**0.06***	0.20
Electronics												
Transformer	12.00	0.56	13.97	12.35	12.39	1.23	1.04	1.06	0.09	0.07	0.01	0.29
NRT	11.8	0.53	14.68	12.28	12.65	1.49	1.26	1.29	0.08	0.05	0.01	0.47
Att2Seq	9.97	0.49	12.93	10.35	10.76	1.28	1.00	1.05	0.23	0.05	0.01	0.20
PETER	11.62	0.60	14.33	12.24	12.39	1.48	1.25	1.26	0.22	0.09	0.03	0.21
PEPLER	12.71	0.59	13.67	13.34	12.75	1.40	1.32	1.26	0.43	0.09	0.03	0.19
P-PEG (ours)	**12.83***	**0.63****	**15.52****	**13.50****	**13.28****	**1.51***	**1.44***	**1.32***	**0.78****	**0.11****	**0.04***	**0.16****
CD and Vinyl												
Transformer	8.24	0.57	12.67	9.12	9.24	1.39	1.03	1.09	0.09	0.07	0.03	0.35
NRT	8.22	0.49	13.30	9.06	10.10	1.43	1.01	1.10	0.08	0.07	0.01	0.33
Att2Seq	7.82	0.45	12.86	8.67	9.71	1.33	0.94	1.02	0.07	0.07	0.02	0.29
PETER	8.83	0.61	13.12	9.63	10.44	1.44	1.10	1.15	0.15	0.08	0.09	0.25
PEPLER	9.73	0.62	12.62	10.13	10.58	1.33	1.19	1.10	0.41	0.08	0.08	0.21
P-PEG (ours)	**9.85****	**0.73****	**13.52****	**10.75****	**10.92****	**1.48***	**1.27****	**1.23****	**0.74****	**0.09***	**0.14****	**0.19***

The best results are shown in bold. B1 and B4 represent BLEU-1 and BLEU-4. R1-P, R1-R, R1-F, R2-P, R2-R, and R2-F denote the precision, recall, and F1-score of ROUGE-1 and ROUGE-2. * and ** indicate the model outperforms all baselines significantly with paired t-test respectively for $p < 0.05$ and $p < 0.01$.

Table 6. The BERTScore Performance of the Explanation Generation Task on the Three Datasets

Model	Cellphones			Electronics			CD		
	BERTS-P	BERTS-R	BERTS-F	BERTS-P	BERTS-R	BERTS-F	BERTS-P	BERTS-R	BERTS-F
Transformer	42.03	39.72	40.32	42.40	41.51	41.65	40.49	38.36	39.20
NRT	40.58	39.11	39.34	42.26	41.48	41.51	41.18	39.06	40.15
Att2Seq	42.31	40.11	40.77	41.48	40.62	40.6	41.11	38.76	39.95
PETER	42.79	39.54	41.18	42.81	42.07	42.22	41.36	39.49	40.20
PEPLER	43.05	41.02	41.88	43.05	42.42	42.52	41.31	39.70	40.30
P-PEG (ours)	**43.64***	**41.53****	**42.31****	**44.54***	**43.41***	**43.73****	**42.38****	**40.04***	**40.93****

The best results are shown in bold. * and ** indicate the model outperforms all baselines significantly with paired t-test, respectively, for $p < 0.05$ and $p < 0.01$.

model can produce more diverse sentences with more feature words because we not only use the GPT model which contains rich language knowledge, but also integrate the semantic information of the text and knowledge into our prompt generator. Besides, the prompt learning that integrates the prompts into each layer of GPT-2 can enhance the understanding ability of our model and generate human-understandable explanations.

6.3 Ablation Study

In this section, we construct a series of ablation experiments to demonstrate the effectiveness of the main parts in our P-PEG. These models are as follows: (1) *w/o. Intent.* We remove the search semantic

Table 7. Performance (NDCG@10) of Ablation Models on the Search Task

Model	**Cellphones**	**Electronics**	**CD and Vinyl**
P-PEG	**0.337****	**0.432****	**0.109****
w/o KG	0.313	0.401	0.089
w/o text	0.317	0.406	0.101
w/o fusion	0.282	0.397	0.086
w/o prompt (GPT fix)	0.233	0.360	0.038
w/o prompt (GPT tune)	0.245	0.393	0.091
P-PEG (Llama2-Chat)	**0.348**	**0.466**	**0.124**

The best results are shown in bold. * and ** indicate the model outperforms all ablation models with removed components significantly, with paired *t*-test respectively for $p < 0.05$ and $p < 0.01$.

signal vector $\mathbf{h}_s$ from the GPT-2 in Equation (10). (2) *w/o. Text.* We remove the text-based encoder in Equation (1) and only use the knowledge-based representations for the search and explanation task. (3) *w/o. Knowledge.* We remove the knowledge-based encoder in Equation (2) and only use the text-based representations for the search and explanation task. (4) *w/o. Prompt.* We remove the prompts $\mathbf{P}_{\text{sch}}$ and $\mathbf{P}_{\text{exp}}$ in Equations (6) and (9) from each layer of GPT-2 respectively, and input the fused representation directly into GPT-2 to observe performance in the case of fixed and fine-tuned GPT-2. (5) *w/o. Fusion.* We remove the fusion unit in Equation (5) and use the concatenation of text-based and knowledge-based representation as input directly. (6) *Llama2-Chat.* To explore the impact of different base models, in addition to using the GPT-2-base model with 124 M parameters in the main experiment results, we also conducted experiments using the Llama2-Chat model [43] with 7B parameters as the base model.

The results of the ablation study on the three datasets are shown in Tables 7 and 8. We can see that these ablation models perform worse than our P-PEG model. In particular, (1) The model *w/o. intent* has degraded performance which shows that the results of the search component can greatly promote the generation of explanations. It also shows the effectiveness of the design that we use the search component to help obtain the explanation. (2) The model *w/o. prompt* that does not use prompt causes the decline of performance in both search and explain generation tasks. This result infers that the prompts can effectively help the model to understand the user's needs and interests, and help the fixed PLM adjust the representation of each layer with fewer parameters to produce products or explanations that meet the user's requirements more. (3) The low result of *w/o. text* and *w/o. knowledge* reveals that both text-based representation and knowledge-based representation can help to capture users' interests and search intents, and directly using one of the representations will lead to inaccuracy. (4) The performance of *w/o. fusion* decrease a lot shows that the gated-based fusion in Equation (5) can effectively take advantage of the two representations and combine their benefits to find suitable products and generate explanations that meet user interests. (5) The performance of *Llama2-Chat* as base model achieves better performance than GPT2-base as the base model, which reveals that the model with a larger base model can achieve better results in both the product search task and explanation generation task. It can retrieve products more accurately and generate explanations with higher text quality and explainability.

6.4 Parameter Sensitivity

There are two important hyper-parameters $\{l, l'\}$ in our model, i.e., the number of search prompts $\mathbf{P}_{\text{sch}}$ and explanation prompts $\mathbf{P}_{\text{exp}}$, which is used in our two tasks respectively. In order to analyze

Table 8. Performance (BLEU-4, F1 Score of ROUGE-2, USR, FMR, and DIV) of Ablation Models on the Explanation Generation Task

Model	B4↑	R2-F↑	USR↑	FMR↑	DIV↓
Cellphones and Accessories					
P-PEG	**0.72***	**1.41***	**0.89***	**0.12**	**0.20***
w/o intent	0.43	1.07	0.82	0.11	0.22
w/o KG	0.48	1.10	0.85	0.09	0.24
w/o text	0.62	1.23	0.84	0.11	0.23
w/o fusion	0.58	1.09	0.84	0.09	0.26
w/o prompt (GPT fix)	0.34	0.98	0.15	0.05	0.75
w/o prompt (GPT tune)	0.45	0.96	0.41	0.09	0.22
P-PEG (Llama2-Chat)	**0.86**	**1.43**	**0.89**	**0.13**	**0.19**
Electronics					
P-PEG	**0.63***	**1.32**	**0.78****	**0.11***	**0.16***
w/o intent	0.59	1.29	0.56	0.10	0.19
w/o KG	0.59	1.30	0.74	0.10	0.21
w/o text	0.61	1.21	0.75	0.09	0.18
w/o fusion	0.60	1.24	0.73	0.08	0.20
w/o prompt (GPT fix)	0.40	1.09	0.08	0.04	0.50
w/o prompt (GPT tune)	0.50	1.01	0.39	0.09	0.16
P-PEG (Llama2-Chat)	**0.68**	**1.35**	**0.79**	**0.11**	**0.15**
CD and Vinyl					
P-PEG	**0.73***	**1.23***	**0.74****	**0.09***	**0.19***
w/o intent	0.70	1.21	0.71	0.08	0.20
w/o KG	0.68	1.15	0.71	0.08	0.21
w/o text	0.70	1.20	0.70	0.08	0.20
w/o fusion	0.67	1.19	0.68	0.07	0.23
w/o prompt (GPT fix)	0.27	1.03	0.03	0.06	0.87
w/o prompt (GPT tune)	0.69	1.16	0.58	0.08	0.24
P-PEG (Llama2-Chat)	**0.76**	**1.27**	**0.76**	**0.10**	**0.18**

The best results are shown in bold. * and ** indicate the model outperforms all ablation models with removed components significantly, with paired *t*-test respectively for $p < 0.05$ and $p < 0.01$.

the sensitivity of our model to the parameters, we evaluate the search task on MAP and NDCG, and the explanation task on BLEU-4, F1 score of ROUGE2, FMR, and DIV under different $\{l, l'\}$.

For the search task, we fix the other parameters and report the results for each metric on all datasets when l is in {5,10,15, 20,25,30,35,40,45,50}, respectively. The results are shown in Figure 3. The results show that the performance is best when the number of search prompts is about 10 for the dataset Cellphones and Electronics, and 40 for the dataset CD and Vinyl. The different optimal number of search prompts may be related to the relevant items per user-query pair for different datasets. The more relevant items per user-query pair reflect that there are more different products for the model to be distinguished within a u-q pair, which needs more search prompts to capture valuable information on user preferences for more accurate search. Moreover, fewer prompts will lead to less information that can be stored and adjusted by each layer of GPT, and too many prompts will lead to information redundancy and not easy to fit the search and explanation task. For the explanation generation task, we evaluate the performance when the number of explanation

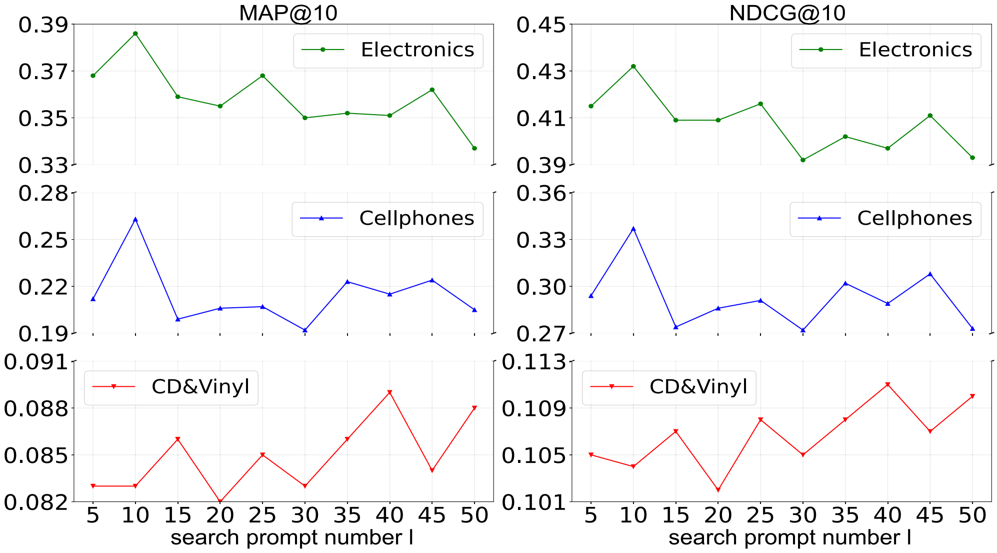

Fig. 3. The performance of P-PEG on the search task for different numbers of search-based prompts.

prompts l' is in {5,10,15,20,25,30,35,40,50}, respectively. The result is illustrated in Figure 4. When the number of explanation prompts is small, the BLEU, ROUGE, and FMR metrics reflect the low quality of explanation generation and low feature matching, and the DIV results suggest that the diversity of explanation generation is bad because the prompts cannot contain enough information to distinguish different users' search intents and preferences. When the number of prompts is too large, the quality of text and the matching degree of features are reduced to a large extent, it is disadvantageous to extract the semantics of the user's feature and explanation. After many experiments, it is found that for these three data sets, when the number of explanation-based prompts equals 20, the model can achieve better results.

6.5 Case Study

To demonstrate the explanation effectiveness of our model intuitively, we conduct a case study and present three explanation examples generated by baselines and our model on the three datasets in Table 9. In the first case from the Electronics dataset, the ground-truth explanation describes a charger that has a fast speed of charging, with the key feature words *charge* and *fast*. First of all, we can note that all of these models produce fluent and readable statements, indicating that their text quality is acceptable, and confirming the fact that these models have higher BLEU and ROUGE. However, the generated explanation from Transformer, NRT, and Att2Seq are too vague and contain no feature words, which is inconsistent with the semantics of ground truth, it also shows the necessity of using explainability metrics such as FMR and USR. On the other hand, PETER and PEPLER generate part of the keyword "charge", but don't get the semantics of ground-truth sentences. Our P-PEG model not only generates the keyword "charge", but also correctly expresses the semantics of fast charging, which is closer to the ground-truth text. In the second case, we can see that the baseline models only mention the word "Sound", and NRT and Att2Seq even generate repeated sentences. Compared with them, our model captures the key feature word "Sound Quality" precisely, which is closest to the ground-truth statement. In the third case, the keywords are "album" and "Christmas". The results from the Transformer and NRT are too broad, and PETER and PEPLER only mention the word "album", but not the word "Christmas" and the semantics are inconsistent.

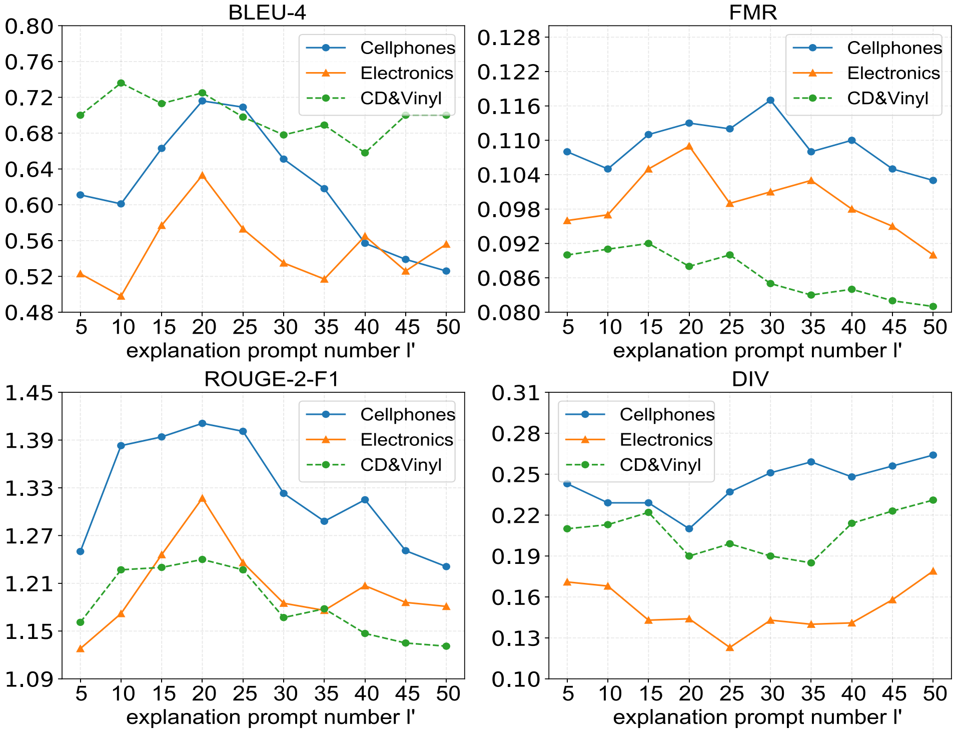

Fig. 4. The performance of P-PEG on the explanation task for different numbers of explanation-based prompts.

Our P-PEG accurately describes the two keywords “Christmas” and “albums”, and makes a positive evaluation of the product.

6.6 User Study

To compare the explanations generated by P-PEG and the baseline models more fairly, we conduct a user study to evaluate the generated sentences. By investigating some of the previous evaluation work [2], we mainly consider three aspects to compare the quality of the sentences. (1) Informativeness. The amount of information available in the generated sentences is an important metric to evaluate the quality of the generated sentences. It is not beneficial to explain the purchase behavior by only generating vague or common sentences. (2) Usefulness. The generated explanation should contain the correct explanation information in line with the user’s purchasing behavior, and can make an effective explanation based on the user’s preferences and user’s needs. (3) Satisfaction. The generated explanations should enable users to achieve a more satisfying shopping experience. In this set of experiments, we compare the explanations generated by our P-PEG model and the state-of-the-art baseline model PEPLER and make a pairwise comparison experiment, which has been proven to be an effective and reliable evaluation method [16]. To better reflect and simulate users’ feedback, we use human evaluation and employ some workers to compare the quality of the explanations generated by the two methods based on the analyzed preferences of users. We hired 25 participants who come from the fields of computer science and artificial intelligence at various universities and research institutions and hold bachelor’s degrees to evaluate the results. We make sure that the workers are not the authors of our article and are familiar with Amazon and experienced in product search and explanation generation research.

Table 9. Three Different Cases of Explanations Generated by Different Methods on Three Datasets

Ground-truth	This thing is **fast** and will **charge** it up from dead in less than 4 hours for me
Transformer	I am very happy with this purchase
NRT	This is a great product
Att2Seq	I have a Kindle and it works great
PETER	I love the fact that it **charges** my Kindle Fire HD and it is very great
PEPLER	This charger is great for **charging** the Kindle Fire HD
P-PEG	Amazon at about $20 for 3 hours and it's **fast** enough to **charge** up
Ground-truth	The **sound quality** is acceptable enough
Transformer	The sound is great
NRT	The sound is good and the sound is good
Att2Seq	The sound is clear and the sound is good
PETER	The sound is great enough
PEPLER	the battery life is great
P-PEG	The **sound quality** is better than I expected
Ground-truth	my mother bought this **album** and played it every Christmas
Transformer	This is a great collection of songs
NRT	The **album** is a great collection of songs
Att2Seq	This is a great collection of songs from the <unk>
PETER	This is one of my favorite **albums** and this is one of my favorite albums
PEPLER	This **album** is a must for any fan of the late great
P-PEG	This is one of the best **Christmas albums** of all time

The key feature words in ground-truth and matched feature words in the generated explanations are shown in bold.

For the dataset, we select the CD and Vinyl as our crowdsourcing dataset. It is a famous and commonly used subset of Amazon. We randomly choose 200 samples from the dataset, where each sample represents a purchase behavior, consisting of user, query, item, history behaviors, and generated explanations. In this experiment, the workers should determine which of the two explanations can better explain the user's purchase behavior based on the user's historical interests, the submitted query, and the ground-truth explanation. To ensure the accuracy of the evaluation, we asked the staff members to vote on the results to determine which explanation performs better. To be fair, we omit the model names for each explanation and randomly mark them as "Models A" and "Model B" for evaluation. These staff members can have three choices on the metric of informativeness, usefulness, and satisfaction based on the known information: Model A or Model B performs better, or they have the same performance. The results of this experiment are shown in Table 10. In general, we can find that on three metrics, the staff members prefer the explanation generated by P-PEG and it can produce more effective explanations to meet the requirements of users. The sentence generated by P-PEG is highly evaluated on the informativeness metric, which is due to the effective combination and application of the text information and the knowledge in the KG, and an effective use of the knowledge in PLM. For the higher performance on satisfaction metric, compared with PEPLER, P-PEG makes use of the information of submitted query and historical behaviors effectively, so that it has a more accurate grasp of user's needs and interests. P-PEG also achieves better results in the metric of usefulness, but not as high as the other two metrics. To report the agreement between the workers for this task, we conducted experiments of the Fleiss' Kappa consistency test [48] and found that the Fleiss' Kappa values of the field

Table 10. Crowdsourcing Results for Explanation Evaluation

Model	**Informativeness**	**Usefulness**	**Satisfaction**
Equal	12%	20%	16%
PEPLER	32%	36%	32%
P-PEG	56%	44%	52%

informativeness, usefulness, and satisfaction are around 0.76, 0.70, and 0.78 respectively. The result indicates a substantial agreement by the workers.

7 Conclusion

In this article, we propose a unified prompt-aware framework P-PEG for personalized search and explanation generation, which can both retrieve suitable products and provide a natural language-based explanation of the retrieved products in a unified manner. To achieve it, P-PEG designs the task-specific prompt generators to better inject the task knowledge into the based PLM. In the search task, the generated search-specific prompts are utilized to help the base PLM understand the user's search intents for the accurate search. In the explanation task, the generated explanation-specific prompts and search results are combined as input to the base PLM, enabling the generation of human-understandable explanations. By incorporating the search results into the explanation task, the interaction between the two tasks is enhanced, and the disparity between search results and explanation generation is reduced. Experimental results have shown that P-PEG outperforms the baseline models in the explanation generation task of the three datasets and the search task of the Electronics dataset, and achieves comparable performance in the search task of the Cellphones and Accessories and CD and Vinyl datasets.

Acknowledgments

The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE, and Engineering Research Center of Intelligent Technology and Educational Application, MOE.

References

[1] Qingyao Ai, Daniel N. Hill, S. V. N. Vishwanathan, and W. Bruce Croft. 2019. A zero attention model for personalized product search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, New York, NY, 379–388. DOI: https://doi.org/10.1145/3357384.3357980

[2] Qingyao Ai and Lakshmi Narayanan R. 2021. Model-agnostic vs. model-intrinsic interpretability for explainable product search. In *Proceedings of the 30th ACM International Conference on Information Amp; Knowledge Management*. ACM, New York, NY, 5–15. DOI: https://doi.org/10.1145/3459637.3482276

[3] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W. Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 645–654. DOI: https://doi.org/10.1145/3077136.3080813

[4] Qingyao Ai, Yongfeng Zhang, Keping Bi, and W. Bruce Croft. 2019. Explainable product search with a dynamic relation embedding model. *ACM Trans. Inf. Syst.* 38, 1 (2019), 1–29. DOI: https://doi.org/10.1145/3361738

[5] Keping Bi, Qingyao Ai, and W. Bruce Croft. 2020. A transformer-based embedding model for personalized product search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 1521–1524. DOI: https://doi.org/10.1145/3397271.3401192

[6] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2787–2795.

[7] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence.* AAAI Press, 301–306.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are Few-Shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Curran Associates, Inc.,1877–1901.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '19)*. Association for Computational Linguistics, 4171–4186. DOI: https://doi.org/10.18653/v1/n19-1423

[10] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL '17)*. Association for Computational Linguistics, 623–632. DOI: https://doi.org/10.18653/v1/e17-1059

[11] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 3816–3830.

[12] Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 8410–8423.

[13] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. 2019. Attentive long short-term preference modeling for personalized product search. *ACM Trans. Inf. Syst*. 37, 2 (2019), 1–27. DOI: https://doi.org/10.1145/3295822

[14] Richard A. Harshman and Margaret E. Lundy. 1994. PARAFAC: Parallel factor analysis. *Comput. Stat. Data Anal*. 18, 1 (1994), 39–72.

[15] Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2225–2240.

[16] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately interpreting click-through data as implicit feedback. *ACM SIGIR Forum* 51, 1 (2017), 4–11. DOI: https://doi.org/10.1145/3130332.3130334

[17] Lei Li, Yongfeng Zhang, and Li Chen. 2020. Generate neural template explanations for recommendation. In *Proceedings of the 29th ACM International Conference on Information Amp; Knowledge Management*. ACM, New York, NY, 755–764. DOI: https://doi.org/10.1145/3340531.3411992

[18] Lei Li, Yongfeng Zhang, and Li Chen. 2022. Personalized prompt learning for explainable recommendation. arXiv:2202.07371. Retrieved from https://arxiv.org/abs/2202.07371

[19] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 345–354. DOI: https://doi.org/10.1145/3077136.3080822

[20] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing continuous prompts for generation. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 4582–4597.

[21] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M. Blei. 2016. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, 59–66. DOI: https://doi.org/10.1145/2959100.2959182

[22] Soon Chong Johnson Lim, Ying Liu, and Wing Bun Lee. 2010. Multi-facet product information search and retrieval using semantically annotated product family ontology. *Inf. Process. Manag*. 46, 4 (2010), 479–493. DOI: https://doi.org/10.1016/j.ipm.2009.09.001

[23] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81.

[24] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*. AAAI Press, 2181–2187.

[25] Danyang Liu, Jianxun Lian, Zheng Liu, Xiting Wang, Guangzhong Sun, and Xing Xie. 2021. Reinforced anchor knowledge graph generation for news recommendation reasoning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, New York, NY, 1055–1065. DOI: https://doi.org/10.1145/3447548.3467315

[26] Jiongnan Liu, Zhicheng Dou, Qiannan Zhu, and Ji-Rong Wen. 2022. A category-aware multi-interest model for personalized product search. In *Proceedings of the ACM Web Conference 2022*. ACM, New York, NY, 360–368. DOI: https://doi.org/10.1145/3485447.3511964

[27] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv:2110.07602. Retrieved from https://arxiv.org/abs/2110.07602

[28] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. arXiv:2103.10385. Retrieved from https://arxiv.org/abs/2103.10385

[29] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR '19)*. OpenReview.net.

[30] Oscar Mañas, Pau Rodríguez López, Saba Ahmadi, Aida Nematzadeh, Yash Goyal, and Aishwarya Agrawal. 2023. MAPL: Parameter-efficient adaptation of unimodal pre-trained models for vision-language few-shot prompting. In *The 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2515–2540.

[31] Kurt T. Miller, Thomas L. Griffiths, and Michael I. Jordan. 2009. Nonparametric latent feature models for link prediction. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*. Curran Associates, Inc., 1276–1284.

[32] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *The 28th International Conference on Machine Learning*. Omnipress, 809–816.

[33] OpenAI. 2023. GPT-4 technical report. arXiv:2303.08774. Retrieved from https://arxiv.org/abs/2303.08774

[34] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, USA, 311–318. DOI: https://doi.org/10.3115/1073083.1073135

[35] Sung-Jun Park, Dong-Kyu Chae, Hong-Kyun Bae, sumin Park, and Sang-Wook Kim. 2022. Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. ACM, New York, NY, 784–793. DOI: https://doi.org/10.1145/3488560.3498515

[36] Matthew E. Peters, and Dan Lecocq. 2013. Content extraction using diverse feature sets. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, New York, NY, 89–90. DOI: https://doi.org/10.1145/2487788.2487828

[37] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.

[38] S. E. Robertson and S. Walker. 1994. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer-Verlag, Berlin, 232–241.

[39] Bjarne Sievers. 2020. Question answering for comparative questions with GPT-2. In *Working Notes of CLEF 2020 – Conference and Labs of the Evaluation Forum*. CEUR-WS.org.

[40] Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM, New York, NY, 650–658. DOI: https://doi.org/10.1145/1401890.1401969

[41] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 926–934.

[42] Zhixing Tan, Xiangwen Zhang, Shuo Wang, and Yang Liu. 2022. MSP: Multi-stage prompting for making pre-trained language models better translators. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 6131–6142.

[43] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv: 2307.09288. Retrieved from https://arxiv.org/abs/2307.09288

[44] Quoc-Tuan Truong, and Hady Lauw. 2019. Multimodal review generation for recommender systems. In *The World Wide Web Conference*. ACM, New York, NY, 1864–1874. DOI: https://doi.org/10.1145/3308558.3313463

[45] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. ACM, New York, NY, 165–174. DOI: https://doi.org/10.1145/2983323.2983702

[46] Damir Vandic, Flavius Frasincar, and Uzay Kaymak. 2013. Facet selection algorithms for web product search. In *Proceedings of the 22nd ACM International Conference on Information Amp; Knowledge Management*. ACM, New York, NY, 2327–2332. DOI: https://doi.org/10.1145/2505515.2505664

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran associates inc., red hook, NY, USA, 6000–6010.

[48] Anthony J. Viera and Joanne M. Garrett. 2005. Understanding interobserver agreement: The kappa statistic. *Family Medicine* 37, 5 (2005), 360–363.

[49] Keping Bi, Qingyao Ai, and W. Bruce Croft. 2021. Learning a fine-grained review-based transformer model for personalized product search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 123–132. DOI: https://doi.org/10.1145/3404835.3462911

[50] Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP '21)*. Association for Computational Linguistics, 4947–4957. DOI: https://doi.org/10.18653/v1/2021.acl-long.383

[51] Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Daxin Jiang, and Kam-Fai Wong. 2022. RecInDial: A unified framework for conversational recommendation with pretrained language models. In *The 2nd Conference of the Asia-Pacific Chapter of Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 489–500.

[52] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA). ACM, New York, NY, 1929–1937. DOI: https://doi.org/10.1145/3534678.3539382

[53] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. AAAI Press, 1112–1119.

[54] Zhenmin Yang, Yonghao Dong, Jiange Deng, Baocheng Sha, and Tao Xu. 2022. Research on automatic news text summarization technology based on GPT2 model. In *Proceedings of the 2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture* (Manchester, United Kingdom). ACM, New York, NY, 418–423. DOI: https://doi.org/10.1145/3495018.3495091

[55] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *The 8th International Conference on Learning Representations*. OpenReview.net.

[56] Yongfeng Zhang, Haochen Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. 2014. Do users rate or review? Boost phrase-level sentiment labeling with review-level sentiment classification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, New York, NY, 1027–1030. DOI: https://doi.org/10.1145/2600428.2609501

[57] Jun Zhu. 2012. Max-margin nonparametric latent feature models for link prediction. In *The 29th International Conference on Machine Learning*. icml.cc/Omnipress.

Received 7 March 2024; revised 8 November 2024; accepted 16 January 2025