IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 May 2024, accepted 28 May 2024, date of publication 3 June 2024, date of current version 10 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3408252

RESEARCH ARTICLE

Generating Explanations for Explainable Recommendations Using Filter-Enhanced Time-Series Information

YUANPENG QU AND HAJIME NOBUHARA, (Member, IEEE)
University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan

Corresponding author: Yuanpeng Qu (qu@cmu.iit.tsukuba.ac.jp)

This work was supported in part by Japan Science and Technology Agency: Support for Pioneering Research Initiated by the Next Generation (JST-SPRING) Program under Grant JPMJSP2124.

ABSTRACT Generating explanations for recommended items is crucial in recommender systems, as it helps users understand how the recommendations align with their preferences, thereby enhancing user satisfaction. Typically, these explanations are produced using natural language generation. However, existing methods often rely solely on item reviews and IDs, ignoring critical historical user behaviors such as previous purchases and sequences, which are essential for improving the effectiveness of recommendations and user satisfaction. To address this issue, we propose a Transformer-based method designed to generate explanations by leveraging time-series information extracted through Transformer-based sequential recommendation. This approach not only captures the temporal dynamics of user interactions but also assigns linguistic meaning to the relationships between time-series information and recommended items, thereby enriching the explanations for recommended items. Additionally, we designed a filter layer that attenuates the noise in the frequency domain of the time-series information, to maximize the benefits. Extensive experiments on three datasets demonstrated that, in most cases, the proposed method generates explanations that are both reasonable and effective compared to state-of-the-art explanation generation methods. Further experiments and analyses have verified the effectiveness of this approach.

INDEX TERMS Natural language generation, transformer, explainable recommendation, time-series information, sequential recommendation.

I. INTRODUCTION

In the current era of exponential growth in online information, recommendation systems are being actively implemented in various fields such as e-commerce, search engines, video and music websites, and social networks [1]. Recently, the importance of providing explanations for recommendations has grown because it enables users to make informed decisions, enhances system usability, and fosters trust. Researchers have investigated different approaches for providing explanations, including using predefined templates that can be customized to specific contexts [2], [3], generating helpful tips to guide users [4], and generating sentences tailored to individual recommendations automatically [5]. We focused on the last task, which has recently gained significant research attention because of the advancements in natural language generation techniques, including recurrent neural networks (RNNs), Transformer [6], [7], and pre-trained language models [8], [9], [10].

Large-scale pre-trained language models such as the GPT series [8], [11], [12] have seen rapid development and wide industrial application, demonstrating significant performance improvements on explanation generation tasks. However, challenges include the need for massive amounts of training data and computational resources, making it difficult for independent researchers to modify or train these models due to the financial and time investments required. Therefore, a viable research approach is to modify and enhance relatively small unpretrained language models, such as Transformer-based models, and enable them to

The associate editor coordinating the review of this manuscript and approving it for publication was Xianzhi Wang.

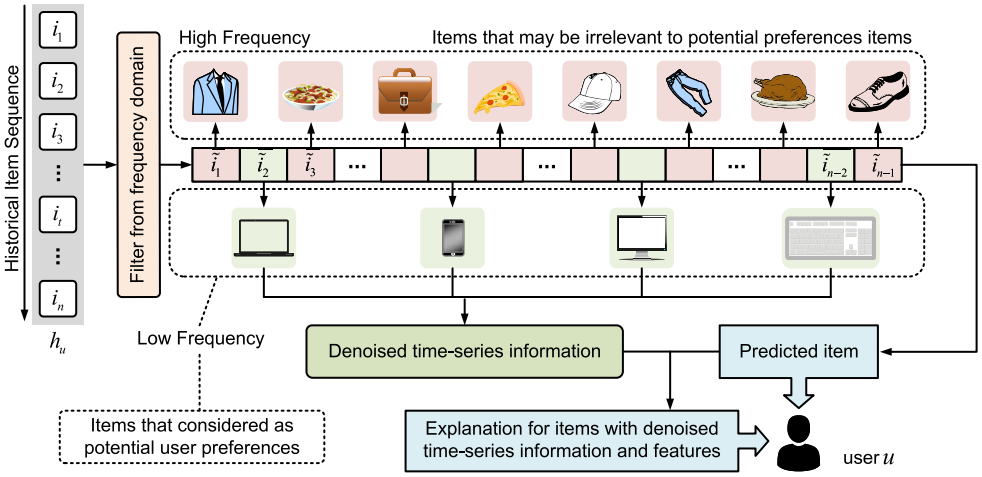

FIGURE 1. Illustration of our motivations. By processing sequential behaviors through the filter layer and the Transformer encoder layer, denoised time-series information is obtained from the historical item sequence at various frequencies. Subsequently, explanations for the predicted item are generated using the modified Transformer layer.

achieve commendable performances. For this approach, there are many existing unpretrained language models. For example, Ni et al. [13] proposed an aspect-conditional masked language model (ACMLM) that uses a fine-tuned BERT to encode features for both the user and the item from an attention layer. Through the prediction of masked tokens, this model can generate diverse sentences. Li et al. [5] introduced a neural template (NETE) explanation generation framework that uses a modified gated recurrent unit (GRU) to learn sentence templates from data and generate template-controlled sentences, providing explanations for specific features. Building on this work, Li et al. [6] proposed a compact, unpretrained Transformer-based model named PETER, which incorporates the use of IDs to predict context words and generate explanations based on them.

However, explanations generated by these methods are related only to the features or reviews of the same item, and they ignore the time-series information, which can easily be obtained from the purchase history of the target user. Studies based on sequential recommendations [14], [15], [16] have shown that the purchase behavior of a user is often closely related to his or her previous purchases, and analyzing the historical behavior sequence of the user allows differentiation between the preferences of that user and those of other users. Therefore, we incorporated time-series information derived from user purchase history as part of the input into the Transformer. This approach enables the generation of explanations that include time-series information, thereby enhancing the trustworthiness and satisfaction of users. Moreover, user behaviors tend to show certain periodic trends. The favorite items of users in their purchase histories tend to be scattered and periodic. Taking Figure 1 as an example, such an interaction is hidden in the purchase record of user u. After buying a display and keyboard, u bought a mouse. If u buys an item with similar characteristics again, a computer-related product could be a suitable recommendation. To generate recommendation reasons for the favorite products of a user more accurately, we applied the Fast Fourier Transform (FFT) to the input sequence of historical purchase records before performing attention learning. This process helped filter out low-frequency information, such as user preferences, and high-frequency information, like unrelated products, thereby reducing noise in the data [17]. We believe that incorporating filter-enhanced time-series information from user purchase sequences can further enable user satisfaction with explainable recommendations.

For these reasons, we designed a Transformer-based unpretrained model to generate explanations using filter enhanced **T**ime-**S**eries **I**nformation when **E**xplainable **R**ecommendations (TSIER) were made. Firstly, we took the user history purchase sequence as the input, after an embedding layer, and we adopted a filter layer that used the FFT to convert the input representations into the frequency domain and an inverse FFT procedure recovered the denoised representations. This is critical for reducing the influence of the noise from unrelated item representations. To implement this approach, we incorporated learnable parameters to encode the input item sequences in the frequency domain, which could be optimized from the backpropagation of the proposed model. After the filter layer, we used the denoised input representation as the input of the Transformer encoder layer for attention learning. This procedure could generate the time-series information based on the denoised input representations and predict the next item user wants to buy through a prediction layer. After that, we combined the time-series information, predicted item, item features obtained from reviews through a sentiment analysis tool [18] at the data preprocess, and item review as the inputs of the Transformer layer to generate the recommendation reason for the predicted

item. Additionally, as the items purchased according to the user history and the review of the predicted item are mostly different in the semantic spaces, it would be problematic to combine them directly as the input of the Transformer, because by doing so, the items purchased according to the user history and predicted item are treated as embedded item features, not words.

To address this issue, we modified the attention mask of the Transformer to generate an explanation for the predicted item, which we called a context prediction task. This task provided linguistic meaning to the relationship between the time-series information of the purchase history sequence of the user and the predicted item and mapped the representation of time-series information onto words. Moreover, we designed the explanation generation task to use the result of the context prediction task, which was included in the features of the time-series information, to generate explanation sentences.

For our experiment, we employed various evaluation methods to assess both the recommendations and the generated reasons, demonstrating the effectiveness of our proposed model. Additionally, we introduced a specialized evaluation metric designed to determine whether the generated sentences incorporate time-series information. The experimental results from three datasets confirmed that, in most cases, our method produces explanations that are both reasonable and effective compared to state-of-the-art explanation generation methods. Further experiments focusing on time-series information and its analysis further validated the effectiveness of our method.

The main contributions of this study can be summarized as follows:

- We proposed the TSIER model to generate recommendation reasons based on the purchase history sequences of users that contain time-series information and to predict items simultaneously, which include more useful features than those in the previous study, as shown in Figure 1.
- We implemented a filter layer that utilizes an FFT to transform the input representations into the frequency domain. Subsequently, an inverse FFT procedure recovers the denoised representations, effectively filtering out unrelated information.
- We bridged the time-series information of the purchase history sequence of the user and the predicted items to assign it a linguistic meaning, thereby ensuring that the generated explanations would be relevant to the predicted item and time-series information.
- We proposed an evaluation metric to measure how time-series features are included in the generated explanations. The experimental results showed that explanations generated by our approach were superior to several strong baselines in terms of evaluation metrics. Further, the generated explanations matched the predicted items and time-series features well.

The rest of this paper is organized as follows: Section II discusses the related work on the generation approaches of explainable recommendations and Transformer-based recommendations. Section III introduces the key notations and concepts before presenting the proposed method. Section IV details the TSIER model design and its technical specifics. The experimental setup and results evaluated using specific metrics, as well as a corresponding analysis, are described in Sections V and VI, respectively. Finally, Section VII provides some concluding remarks.

II. RELATED WORKS

A. EXPLAINABLE RECOMMENDATION

Explanation generation has emerged as an increasingly important task in the field of explainable recommendations [19], [20], which has gained significant attention in both the machine learning and human–computer interaction domains [21], [22], and served as our primary research focus. We adopted a machine learning perspective because of its extensive range of possibilities for designing novel methods to generate explanations for the recommendations. The existing methods for explainable recommendations can be categorized into various types such as ranked sentences [23], [24], predefined templates [2], [25], reasoning rules [26], [27], item features [28], [29], and knowledge graphs [30], [31]. Recently, the generation of explanations for natural language processing has gained popularity with the proposal of the Transformer model. The Transformer model is a deep-learning model based entirely on the self-attention mechanism because it is suitable for parallel computing. The complexity of this model makes it more accurate and superior to previously popular RNN-based networks. However, the existing explanation generation method only generates an explanation based on the Transformer [6] or RNN [5], [32], ignoring the order of user behaviors that play important roles in increasing recommendation accuracy. This study was aimed at addressing this issue.

B. NATRUAL LANGUAGE GENERATION VIA TIME-SERIES INFORMATION

Based on the Transformer model, explanation generation for one target user always includes the personal information of the user, such as the user ID and item ID that the user wishes to buy. The existing approaches [4], [5], [33] adopt multilayer perceptrons to encode the ID pairs and decode them to the words of a sentence via RNN and Transformer-based models. This method is used for tip generation [33], review generation [4], explanation generation [5], and personalized generation [6]. Further, interest in personalized natural language generation using pre-trained models has been increasing. However, for Transformer-based explainable recommendation tasks, none of these methods use time-series information to predict the recommended items for the target user. The existing studies [5], [6] that have employed the Transformer model for explanation generation have utilized the relevance between the user and item IDs to generate a sentence that can explain its relationship. Inspired by

this work, we bridged the time-series information obtained from the purchase history sequence of the user and the items recommended to the user, assigning them a linguistic meaning and generating explanations based on this meaning.

C. TRANSFORMER-BASED SEQUENTIAL RECOMMENDATION

Collaborative filtering [34], [35] has been extensively utilized in various recommender system models in the early years. These models [36], [37] acquire the preferences of the users based on their past interactions. However, these general recommendation models do not consider the sequence of user actions that play a vital role in the recommendations. Recently, recommendations based on time-series information have been developed using RNN-based tasks such as a gated recurrent unit [38], [39] and long short-term memory (LSTM) [40], which are being increasingly employed to model user behavior sequences. In addition, sequential recommendations using encoder–decoder models have advanced significantly since the Transformer [7] was created from natural language processing architecture, which has resulted in unparalleled outcomes. Examples of Transformer-based sequential recommendation models include SASRec [41], which is a recommendation system that utilizes a self-attention mechanism, and the bidirectional Transformer encoder-based model BERT4Rec [42]. Based on Transformer, our proposed method not only generates explanations, but also makes recommendations. We incorporated a filter layer that employs the FFT and the inverse FFT. This layer effectively filters out irrelevant information, thereby enabling the provision of personalized recommendations more effectively.

III. PRELIMINARIES

A. PROBLEM STATEMENT

The key notations and concepts of the proposed method are shown in Table 1. Assume that we have a set of users $\mathcal{U}$ and items $\mathcal{I}$, where $u \in \mathcal{U}$ represents a user and $i \in \mathcal{I}$ represents an item. Our proposed approach produces an explanatory sentence $E_{u,i} = (e_1, e_2, \ldots, e_w)$ that justifies the recommendation of item i to target user u. In addition, it outputs recommended item i_n^u of user u predicted by the Transformer encoder layer as the recommendation result and generates reason $E_{u,i}$ for the user simultaneously. Here, $\{e_x\}_{x=0}^{w}$ represents the words of the generated sentence, and n and w represent the lengths of the fixed-length purchase history sequence and generated explanation sentence, respectively.

B. FOURIER TRANSFORM

1) DISCRETE FOURIER TRANSFORM (DFT)

The DFT plays a crucial role in digital signal processing. In this study, it was utilized to denoise unrelated information from the input representation. We only considered a one-dimensional (1D) DFT in this work. Given a sequence $\{x_n\}_{n=1}^{N}$, the 1D DFT can convert the original sequence into the sequence of complex numbers in the frequency domain by:

$$X_k = \sum_{n=1}^{N} x_n e^{-\frac{2\pi i}{N} nk}, \quad 1 \le k \le N, \tag{1}$$

where N is the length of the sequence, i is the imaginary unit, $e^{-\frac{2\pi i}{N} nk}$ is the twiddle factor, and X_k is the spectrum of the sequence $\{x_n\}$ at the frequency $\omega_k = 2\pi k/N$. Note that the DFT is a one-to-one transformation in the time and frequency domains. Furthermore, the frequency representation sequence $\{X_k\}_{k=1}^{N}$, can be converted back into the original feature domain using an inverse DFT (IDFT), which is formulated as

$$x_n = \frac{1}{N} \sum_{k=1}^{N} X_k e^{\frac{2\pi i}{N} nk}. \tag{2}$$

2) FAST FOURIER TRANSFORM (FFT)

The FFT is an efficient algorithm for calculating the DFT of a sequence. It exploits the symmetry and periodicity of the term $e^{-\frac{2\pi i}{N} nk}$, reducing the computational complexity of the DFT from $O(N^2)$ to $O(N \log N)$. For the input x_n, its DFT is known to be conjugate-symmetric, implying that the first half of the DFT, $\{X_k\}_{k=0}^{\lceil N/2 \rceil}$ contains complete information about the frequency features of x_n. By performing the IDFT on $\{X_k\}_{k=0}^{\lceil N/2 \rceil}$, a full and real discrete signal is recoverable, showcasing the utility of the FFT and inverse FFT (IFFT). In this paper, we represent FFT and IFFT using $\mathcal{F}$ and $\mathcal{F}^{-1}$, respectively.

TABLE 1. Description of symbols used in the model.

Symbol	Description
φ	training set
$\mathcal{U}$	set of users
$\mathcal{I}$	set of items
F	set of features
R	set of reviews
H	set of user purchase history
h	user purchase history sequence
$\mathbf{I}$	embeddings of items
$\mathbf{i}$	embedding of item i
W, b	learnable weight parameters
T	time-series information
M	modified attention mask
Z	input item representation
Y	learnable filter parameter for the filter layer
E	word sequence of an explanation
c	probability distribution of generated word
LayerNorm$(\cdot)$	Layer normalization function
Dropout$(\cdot)$	dropout operation function
ReLU$(\cdot)$	ReLU activation function
softmax$(\cdot)$	softmax function
$\mathcal{F}(\cdot), \mathcal{F}^{-1}(\cdot)$	Fast Fourier Transform (FFT), inverse FFT

IV. PROPOSED METHOD

In this section, we present the details of the proposed method TSIER. As shown in Figure 2, after the embedding layer, the item sequence embeddings are transformed from the time domain to the frequency domain using the FFT

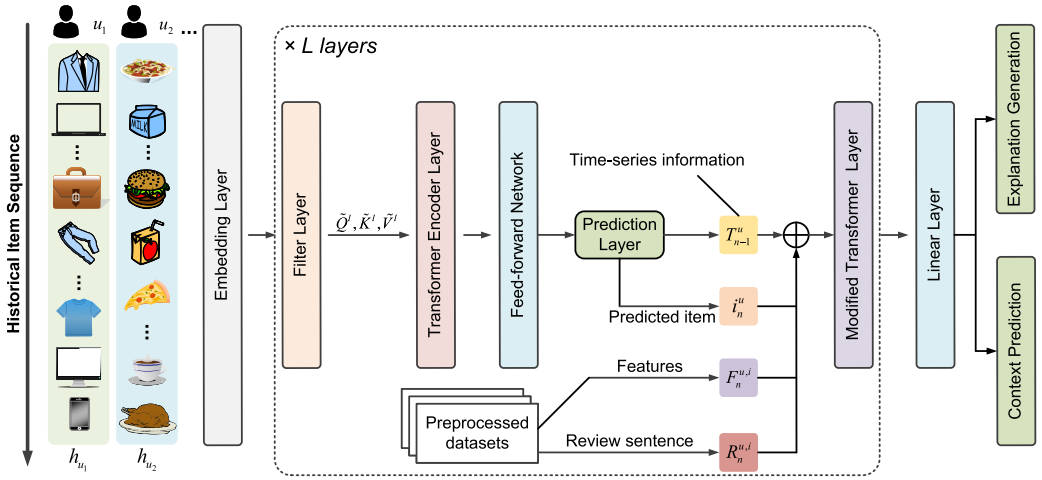

FIGURE 2. Overview of the proposed model. Our model generates explanations for the recommended item using denoised time-series information. Following the item embedding layer, our model incorporates *L* blocks, each containing a Transformer encoder layer with a filter layer and a modified Transformer layer, etc. These are used to predict the next item in the target user's preference list and generate explanations for it.

algorithm mentioned in Section III. Following this, the denoised time-series information and predicted item are output from the Transformer encode layer and combined with the predicted items features and review from the preprocessed dataset as the input representation of the modified Transformer. Finally, the explanation sentence for the predicted item is generated based on the output from the modified Transformer.

A. INPUT REPRESENTATION AND EMBEDDING LAYER

Sequential recommendation focuses on modeling the sequence of user behaviors based on implicit feedback, represented as a list of item IDs in sequential recommendation. We define the item ID set as $\mathcal{I} = (i_1, i_2, \ldots, i_{|\mathcal{I}|})$ and user ID set as $\mathcal{U} = (u_1, u_2, \ldots, u_{|\mathcal{U}|})$, where $i \in \mathcal{I}$ and $u \in \mathcal{U}$ represent an item and a user, respectively. Additionally, the collection of user purchase histories is denoted as $H = \{h_1, h_2, \ldots, h_{|\mathcal{U}|}\}$. For a given user u, their his or her purchase history sequence is $h_u = [i_1^u, i_2^u, \ldots, i_t^u, \ldots, i_n^u]$, where $h_u \in H$, $u \in \mathcal{U}$, and $i_t^u \in \mathcal{I}$. Here, i_t^u signifies the item purchased by user u at time step t, and n is the length of the purchase history sequence. For h_u, it can be embedded as

$$\mathbf{I}_u = \left[\mathbf{i}_1^u, \mathbf{i}_2^u, \ldots, \mathbf{i}_n^u\right], \tag{3}$$

where $\mathbf{i}_t^u$ is the embedding of item i_t^u. In addition, we integrate a learnable position encoding matrix $\mathbf{P}$ that shares the same embedding dimensions as the item embedding. Furthermore, operations such as dropout and layer normalization are also applied:

$$I_u = Dropout(LayerNorm(I + P)). \tag{4}$$

After the Transformer encoder layer, we use T_{n-1}^u and i_n^u to denote the time-series information prior to i_n^u and predicted item i_n^u, respectively. To ensure that the generated explanations reflect the characteristics of the predicted item i_n^u, we concatenate its features $F_n^{u,i} = (f_1, f_2, \ldots, f_{|F|})$, which are extracted from its review $R_n^{u,i} = (r_1, r_2, \ldots, r_w)$ using a sentiment analysis method [18], with review $R_n^{u,i}$ itself, as the input vector of the modified Transformer. Here, w denotes the number of words in the review sentence. Thus, by concatenating all the inputs mentioned above, for a target user u, we obtain the integrated input vector $\mathcal{S} = (i_n^u, T_{n-1}^u, F_n^{u,i}, R_n^{u,i})$ for the modified Transformer. Moreover, we employ positional encoding $P_{\mathcal{S}} = (p_1, \ldots, p_{|\mathcal{S}|})$ to denote the order of the input sequence $\mathcal{S}$. Finally, we derive the final input sequence $\mathcal{S}_0$:

$$\mathcal{S}_0 = (\mathcal{S}_1^0, \mathcal{S}_2^0, \ldots, \mathcal{S}_{|\mathcal{S}|}^0), \tag{5}$$

where $\mathcal{S}_0 \in \mathbb{R}^{d \times |\mathcal{S}|}$, with d and $|\mathcal{S}|$ representing the dimension of the embedding and the length of the input sequence $\mathcal{S}$, respectively.

B. LEARNABLE FILTER-ENHANCED LAYER

Based on the embedding layer, we designed a learnable filter layer for the Transformer encoder layer, aimed at filtering out unrelated information for time-series data, as shown in Figure 3. Within the filer layer, a filtering operation is executed for each feature dimension in the frequency domain, followed by applying a skip connection and layer normalization.

Starting with the input item representation matrix $Z^l \in \mathbb{R}^{n \times d}$ of the l-th layer and setting $Z^0 = I_u$, we perform the FFT is performed along the item dimension to shift Z^l into the frequency domain:

$$X^l \leftarrow \mathcal{F}(Z^l) \in \mathbb{C}^{n \times d}, \tag{6}$$

where $\mathcal{F}(\cdot)$ denotes the 1D FFT, and $X^l \in \mathbb{C}^{n \times d}$ is a complex tensor and representing the spectrum of Z^l. The spectrum is then modulated by multiplying a learnable filter Y, which

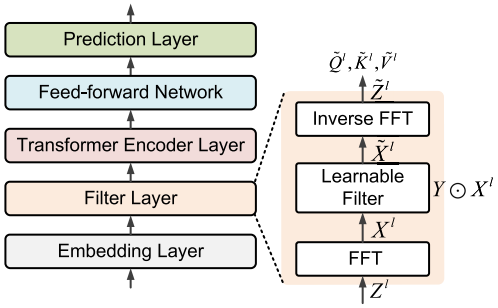

FIGURE 3. Transformer encoder layer, enhanced with a filter layer. By applying the FFT and IFFT along with a learnable filter, we can denoise the input item representation Z^l to $\widetilde{Z}^l$.

can be optimized through stochastic gradient descent to represent an arbitrary filter adaptively in the frequency domain, as illustrated below:

$$\widetilde{X}^l = Y \odot X^l, \tag{7}$$

where $\odot$ signifies the element-wise multiplication. Finally, we utilize the inverse FFT to convert the modulated spectrum $\widetilde{X}^l$ back into the time domain and update the sequence representations:

$$\widetilde{Z}^l \leftarrow \mathcal{F}^{-1}(\widetilde{X}^l) \in \mathbb{R}^{n \times d}, \tag{8}$$

where $\mathcal{F}^{-1}(\cdot)$ represents the inverse 1D FFT, transforming the complex tensor into a real-number tensor. Following the process of this filter layer, we can effectively minimize irrelevant information can be effectively minimized, thereby obtaining more refined time-series information.

C. TRANSFORMER ENCODER LAYER

In this study, we utilized the Transformer encoder layer to predict recommendation item and employed the modified Transformer layer to generate explanations for it. For the embedding $\widetilde{Z}^l$, following the linear projection, we obtained the queries $\widetilde{Q}^l \in \mathbb{R}^{n \times d}$, keys $\widetilde{K}^l \in \mathbb{R}^{n \times d}$, and values $\widetilde{V}^l \in \mathbb{R}^{n \times d}$ as inputs of the Transformer encoder layer, and the self-attention mechanism of the Transformer encoder layer can be described as follows:

$$Attention\left(\widetilde{Q}^l, \widetilde{K}^l, \widetilde{V}^l\right) = softmax\left(\frac{\widetilde{Q}^l \left(\widetilde{K}^l\right)^T}{\sqrt{d}}\right)\widetilde{V}^l, \tag{9}$$

Through this approach, our model is capable of learning the information from the user behavior sequence at the top layer, generating the time-series information, and predicting the item in the prediction layer. Additionally, our proposed model features a multi-headed self-attention mechanism with h heads and L layers. For the i-th head, the process is as follows:

$$\begin{aligned} head_i &= Attention(Q_i, K_i, V_i), \\ MultiHead(Q, K, V) &= Concat(head_1, \ldots, head_h)W, \end{aligned} \tag{10}$$

where Q_i,K_i,$V_i \in \mathbb{R}^{n \times \frac{d}{h}}$, $i \in 1, \ldots, h$ and learnable output matrix $W \in \mathbb{R}^{d \times d}$. Hence, the multi-head attention for our proposal can be described as $MultiHead(\widetilde{Q}^l, \widetilde{K}^l, \widetilde{V}^l)$.

Furthermore, we applied the position-wise feed-forward network (FFN) for $\widetilde{Z}^l$ to make the network non-linear. The FFN is a two-layer network with the ReLU activation function, as shown below:

$$FFN(\widetilde{Z}^l) = ReLU(\widetilde{Z}^l W_1 + b_1)W_2 + b_2, \tag{11}$$

where W_1, $W_2 \in \mathbb{R}^{4 \times d \times d}$,$b_1 \in \mathbb{R}^{4 \times d}$, $b_2 \in \mathbb{R}^{1 \times d}$ are learnable weight parameters.

D. ITEM PREDICTION LAYER

The purchase history sequence of each user is transformed into a fixed-length sequence by either truncating or padding with "$\langle pad \rangle$," ensuring that all training samples have the same length n ($[i_1^u, i_2^u, \ldots, i_t^u, \ldots, i_n^u]$). The network is sequentially trained for each item in the purchase history of u at time step t, and the output at each step is defined as o_t, which can be formulated as follows:

$$o_t = \begin{cases} \langle pad \rangle, & \text{if previous item is a padding item} \\ i_t^u, T_{t-1}^u, & 1 < t < n \\ i_n^u, T_{n-1}^u, & t = n, \end{cases} \tag{12}$$

where T_{t-1}^u represents the time series information of user u from the start time to the time step t, $t \in (1, n-1]$. Further, we predicted the next item that target user u wants to buy is predicted by using the output $\widetilde{Z}^L$ from the Transformer encoder with L layers. A matrix factorization (MF) layer is employed to predict the relevance of the next item i_n^u. This relevance is then transformed into a recommendation probability using the softmax function:

$$\hat{y} = softmax(I_t^\top \widetilde{Z}^L), \tag{13}$$

where $\hat{y}$ represents the relevance of the predicted item i_t^u at time step t for user u, and $I_t \in \mathbb{R}^{n \times d}$ represents the item embedding of the i_t^u. The binary cross-entropy loss is employed as the loss function L_r to achieve a higher score, which is defined as

$$L_r = -\sum_{h_u \in H} \sum_{t \in [1,\ldots,n]} \left(\log(\hat{y}) + \sum_{j \notin h_u} \log\left(1 - \hat{y}_j\right) \right), \tag{14}$$

where $\hat{y}_j$ represents the sample for which one negative instance is randomly generated at each time step t in each sequence.

E. MODIFIED ATTENTION MASK FOR TRANSFORMER LAYER

To generate the explanation with time-series information, we modified the attention mask of the Transformer layer as shown in Figure 4. Note that this Transformer layer is distinct from the one described in the previous section. Recall that the input for the modified Transformer layer

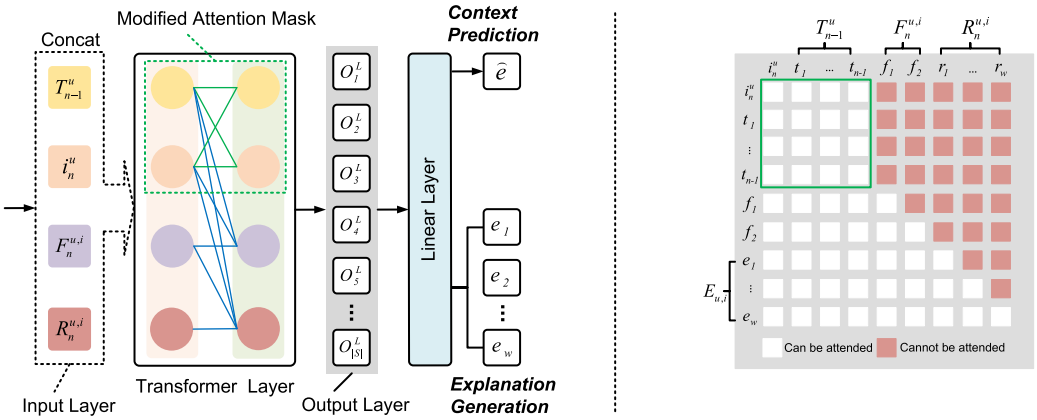

FIGURE 4. Modified Transformer layer (left) with the modified attention mask M (right). The attention mask shown by the green box is modified to ensure that the time-series information can establish a relationship with the representation of the predicted item.

is $\mathcal{S} = (i_n^u, T_{n-1}^u, F_n^{u,i}, R_n^{u,i})$ and $\mathcal{S}_0$ is the input vector $\mathcal{S}$ that includes the positional encoding. Following the linear projection, we can also obtain the $\{Q_\mathcal{S}, K_\mathcal{S}, V_\mathcal{S}\} \in \mathbb{R}^{|\mathcal{S}|\times d}$ can be obtained as the inputs of the modified Transformer, and the modified attention mask MA_s for Transformer is can be computed as

$$MA_\mathcal{S} = softmax(\frac{Q_\mathcal{S}K_\mathcal{S}}{\sqrt{d}} + M)V_\mathcal{S},$$
$$M = \begin{cases} 0, & \text{can be attended} \\ -\infty, & \text{cannot be attended}, \end{cases} \tag{15}$$

where $M \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$. This design ensures that the attention mechanism focuses only on the historical behavior sequence of user u, T_{n-1}^u, and the representation i_n^u of the predicted item in the Transformer layer, while preventing other inputs from attending to each other and allowing them to be referenced from left to right [6]. Thus, besides the first two tokens, T_{n-1}^u and i_n^u can attend to both past and future tokens, whereas other tokens can attend to past tokens and prevent them from attending to future tokens. As in Eq.10, the multi-head attention for modified Transformer can be described as $MultiHead(Q_\mathcal{S}, K_\mathcal{S}, V_\mathcal{S})$. Further, the final output of the modified Transformer model with L layers is

$$O_L = \left(O_1^L, O_2^L, \ldots, O_{|\mathcal{S}|}^L\right), \tag{16}$$

where $|\mathcal{S}|$ represents the length of O_L, this which means that the dimensions of the sequence input $|\mathcal{S}|$ and output O_L are equal. We also employed a linear layer for the final output O_L to distinguish the output representation into context prediction and explanation generation. This layer facilitates the conversion of the final output into a V-size vector, which is given by:

$$c_t = softmax(W_v O_t^L + b_v), \tag{17}$$

where $W_v \in \mathbb{R}^{|V|\times d}$ and $b_v \in \mathbb{R}^{|V|}$ represent learnable weight parameters, c_t denotes the probability distribution of the word in the word dictionary at time step t in the output, and $|V|$ represents the size of the word dictionary.

F. EXPLANATION GENERATION

For the explanation generation task, we utilized the negative log-likelihood (NLL) L_e as the loss function and computed the average NLL loss during training:

$$L_e = -\frac{1}{\varphi}\sum_{(u,i)\in\varphi}\frac{1}{|w|}\sum_{t=1}^{|w|}\log\left(c^{e_t}_{n+\left|F_n^{u,i}\right|}\right), \tag{18}$$

where φ represents the training set and $|w|$ represents the length of the input review sentence $R_n^{u,i}$. Further, $c^{e_t}_{n+|F_n^{u,i}|}$ represents the probability of a word being generated for time step t; the generation starts from position $n + |F_n^{u,i}|$ in the output. In the testing stage, the input of the model, $R_n^{u,i}$, is replaced with "$\langle$ bos$\rangle$" and the probability of the next word is obtained from $c_{\langle bos\rangle}$. Moreover, we decode the words with the largest probability and concatenate the next word at the end of the previous word in sequence to generate a complete sentence with $|w|$ length until the "$\langle$ eos$\rangle$" sign appears.

G. CONTEXT PREDICTION

If the proposed model includes only an explanation generation task, it can be challenging for the Transformer to match the features of the time-series information and predicted items, which results in the generation of repetitive words. Therefore, we included a context prediction task to address the difficulty of generating meaningful explanations when the Transformer had access to only the time-series information and predicted items. By mapping these inputs to explanations and creating connections between them, the context prediction task facilitates feature matching and generates more coherent explanations. The input data parameters (i_n^u and T_{n-1}^u) are bidirectionally accessible. Therefore, it is possible to generate context that includes these features. NLL was applied as the loss function for this task, and it is defined as:

$$L_e = -\frac{1}{\varphi}\sum_{(u,i)\in\varphi}\frac{1}{|w|}\sum_{t=1}^{|w|}\log\left(c^{e_t}_{|n|}\right). \tag{19}$$

In contrast to L_e, all predicted words in this formula are derived from the n-th position of the output layer.

TABLE 2. Details of the experimental datasets after data preprocessing.

	Yelp	Amazon Movie	TripAdvisor
#user	27,147	7,506	9,765
#item	20,266	7,360	6,280
#feature	7,340	5,389	5,069
#total	1,293,247	442,783	320,023
#avg sequence length	63.81	60.02	50.96
#max sequence length	106	85	67

H. MULTITASK LEARNING

After computing the loss for each task, the multitask learning loss function is evaluated as

$$L = \min_{\theta} \{L_r + \omega(L_e + L_c)\}, \tag{20}$$

where θ represents all trainable parameters in the model. Given that the tasks of the modified Transformer serve as a regularization term, the regularization coefficient ω can be manipulated to control the learning process of the explanation generation task.

V. EXPERIMENTAL SETUP

In this section, we initially provide a brief overview of the experimental settings. Subsequently, we present extensive experiments to assess the effectiveness of our proposed model by addressing the following research questions:

- RQ1: How does TSIER perform compared with the state-of-the-art (SOTA) explanation generation models?
- RQ2: Do the explanations generated by TSIER effectively incorporate time-series information?
- RQ3: How does the filter-enhanced Transformer layer from TSIER perform compared to other SOTA sequential recommendation models?
- RQ4: How do the various components of TSIER impact its overall performance?
- RQ5: What is the effect of different hyperparameters on the performance of TSIER?

A. DATASET

We concentrated on analyzing the dataset by examining one user and all items he or she had reviewed, with the specifics of the dataset presented in Table 2. For this purpose, a sentiment analysis method [18] was utilized to extract features from each review, and the maximum sentence length was restricted to 100 characters. The purchase history of the user was represented as a sequence of items, where each item was accompanied by a review written by the user and several extracted features. When n was the length of the sequence, the first $n-2$ items in the sequence were used for training; the penultimate and final items were used for validation and testing, respectively. The proposed model was applied to Yelp, TripAdvisor, and Amazon Movie datasets. The data were split into training, testing, and validation sets in the ratio $(n-2):1:1$.

B. EVALUATION METRICS

We measured the generated explanation from two perspectives: text quality and explainability. For the former, we utilized BLEU [43] to assess machine translation and ROUGE [44] to evaluate text summarization for gauging the accuracy and comprehensibility of the generated explanation. We used BLEU-1 and BLEU-4 for machine translation and Recall, Precision, and F1 for ROUGE-1 and ROUGE-2, respectively. Our objective was to explain the recommendation outcomes; however, relying solely on the above-mentioned metrics may not be sufficient to evaluate whether the purchase history sequence and review features of a user are matched appropriately. For example, many sentences generated by the baselines and the proposed method may be exactly the same; this situation may not be appropriate to explain the unique properties of different recommendation results well. To address this issue, we utilized the unique sentence ratio (USR) to determine the percentage of difference between each generated sentence [5]:

$$USR = \frac{\rho}{\mathcal{T}}, \tag{21}$$

where ρ represents the set of unique sentences generated from baselines or proposed model, and $\mathcal{T}$ is the total number of testing samples.

Furthermore, for explainability, we employed three evaluation metrics proposed by Li et al. [5] (feature matching ratio (FMR), feature coverage ratio (FCR), and feature diversity (DIV)) to assess the feature matching accuracy. FMR determines whether the explanation generated by the proposal contains features of the ground truth:

$$FMR = \frac{1}{\mathcal{T}} \sum_{(f_{u,i}, \widehat{E}_{u,i}) \in \mathcal{T}} \delta\left(f_{u,i}, \widehat{E}_{u,i}\right),$$
$$\delta(f_{u,i}, \widehat{E}_{u,i}) = \begin{cases} 1, & \text{if } f_{u,i} \in \widehat{E}_{u,i} \\ 0, & \text{otherwise,} \end{cases} \tag{22}$$

where $f_{u,i}$ and $\widehat{E}_{u,i}$ represent the ground-truth feature of user u, and the explanation sentence generated by the baselines and proposed model, respectively. FCR computes the proportion of distinct features included in all generated explanations to the total number of features in the entire dataset:

$$FCR = \frac{\tau}{F_{u,i}}, \tag{23}$$

where τ represents the features included in all generated explanation sentences, and $F_{u,i}$ is the set of all features found in the ground-truth explanations. DIV measures the percentage of identical features between any two generated explanations:

$$DIV = \frac{2}{\mathcal{T}(\mathcal{T}-1)} \sum_{(u,u',i,i') \in \mathcal{T}} \left|\widehat{F}_{u,i} \cap \widehat{F}_{u',i'}\right|, \tag{24}$$

where $\widehat{F}_{u,i}$ and $\widehat{F}_{u',i'}$ represent two feature sets of two generated explanation sentences and $|\cdot|$ denotes the number of features in the resulting set.

To the best of our knowledge, no methods exist for evaluating whether generated sentences contain time-series information. Therefore, we proposed an evaluation metric TFMR@K to observe the effect of the sentences generated via time-series information more intuitively. This metric refers to the ratio of all words that contain time-series information features in the generated sentence matching the latest K items in the purchase history sequence of the user. For items with multiple words, such as "washing machine," the generated sentence must contain all words of the item; otherwise, it is not counted. TFMR@K is defined as

$$TFMR@K = \frac{1}{\mathcal{T}} \sum_{(e_t, I_u^K) \in \mathcal{T}} \frac{1}{K} \delta\left(e_t, h_u^K\right),$$
$$\delta(e_t, h_u^K) = \begin{cases} 1, & \text{if } e_t \in h_u^K \\ 0, & \text{otherwise,} \end{cases} \tag{25}$$

where h_u^K, and e_t represent the latest K item set in the purchase history sequence of user u and a word from the sentence generated by the proposed method containing time-series information features, respectively. In this study, the time-series information feature of an item was its name. Our model also can make a recommendation from the Transformer with filter layer. For the evaluation metrics, we utilized two standard top-N metrics to assess the effectiveness of the proposed approach, i.e., hit ratio (HR@N) and normalized discounted cumulative gain (NDCG@N). Further, to prevent the generation of redundant recommendation results and ensure evaluation accuracy, we combined the ground truth data with 100 randomly generated negative samples, ranked them alongside the ground truth items, and let N = 5 and 10. This approach enabled the calculation of the NDCG@N and HR@N scores.

C. COMPARISON OF METHODS

The following SOTA models were used as baselines to compare the performance of the proposed model. These baselines included Transformer, BERT, LSTM, and GRU. All baselines parameters were trained together.

- PETER+ [6] is an explanation-generating model that uses Transformer. We used it to generate an explanation from the user ID, item ID, features, and reviews for comparison with the effectiveness of the proposed method.
- ACMLM [13] is a modified BERT [9] model that encodes the features of the user and item using tokens masked from BERT, which can generate an explanation sentence for it.
- Att2Seq is a review generation approach with a two-layer LSTM [40]. We took the explanations as reviews and removed the attention module because it made the generated content unreadable.
- NETE [5] is a customized GRU that integrates a specified feature into the decoding process to produce structured explanations like those of the templates. This model can also provide recommendations.
- TSIER-GRU (T-GRU) replaces the Transformer layer of the proposed method with a unidirectional GRU. It also performs context and rating prediction.

Our proposed model can also be compared in performance with other sequential recommendation models. We compared our methods with two types of representative sequential recommendation models: non-sequential models (MF) and sequential models (Transformer, BERT, GRU):

- BPR-MF [16] represents a classic non-sequential method for learning personalized ranking from implicit feedback. It optimizes matrix factorization using a pairwise Bayesian personalized ranking (BPR) loss.
- SASRec [41] predicts items using the self-attention mechanism, utilizes a mask to mask the ground truth item, and employs the model to predict the masking results.
- SSE-PT [45] incorporates personalized embeddings to enhance the performance of the Transformer model in the context of sequential recommendations.
- BERT4Rec [42] utilizes deep bidirectional Transformer encoders to model user behavior sequences and introduces the Cloze task. This task involves predicting masked items by considering both the left and right context within the sequences.
- GRU4Rec [21] is a method that employs RNNs to model user action sequences for session-based recommendations. In this approach, the feedback sequence of each user is treated as an individual session.

D. IMPLEMENTATION DETAILS

The data preprocessing for our approach involved extracting words from the three datasets to construct a dictionary utilizing the 30,000 most frequently used words as word embeddings $|V|$. For baselines, all hyperparameters were set following the suggestions from the original papers. We implemented all baselines and the proposed method in Python using PyTorch. For our approach, we adopted consistent parameter settings: an embedding dimensionality d of 512, a number of modified Transformer layers L of 3, a Transformer encoder layer with filter layer count L_f of 3, and a feedforward network dimensionality of $4d = 2048$. For training, we set the batch size to 36, the number of features extracted from reviews in the inputs to 2, and the length of the purchase history sequence n to 50. Top-N sampling ($N = 15$) [46] was used for word sampling in the context prediction task. The maximum explanation length was set to 25, as the average length of the ground truth sentences was approximately 24. Thus, the input size length of the modified Transformer layer $|S|$ is $(1+2+25+50 = 78)$. The learning rate decreased by a factor of 0.25. The training was stopped after five iterations when the learning rate decreased, and the saved model was used for prediction and generation.

E. PARAMETER SIZE OF THE TSIER

For the parameter size of the proposed model TSIER, we calculate it based on the hyperparameter settings

TABLE 3. Explanation generation results for text quality on three datasets. The experiments were conducted five times, with the mean and standard deviation reported.

	Text Quality							
	BLEU-1	BLEU-4	ROUGE-1,r	ROUGE-1,p	ROUGE-1,f	ROUGE-2,r	ROUGE-2,p	ROUGE-2,f
	Yelp							
Att2Seq	10.63 ±0.12	0.59±0.03	11.43±0.07	18.41±0.15	13.30±0.08	1.25±0.06	1.76±0.09	1.37±0.05
ACMLM	7.11±0.09	0.33±0.02	9.09±0.07	8.08±0.11	7.12±0.10	0.64±0.03	0.61±0.02	0.55±0.04
NETE	21.93±0.17	2.96±0.05	20.03±0.19	30.01±0.24	23.88±0.21	5.19±0.06	8.25±0.12	6.01±0.06
PETER+	23.04±0.13	3.42±0.04	**25.83**±0.26	34.03±0.11	26.63±0.32	7.15±0.11	9.50±0.20	7.75±0.14
T-GRU	23.42±0.16	3.46±0.10	23.18±0.20	30.12±0.27	24.53±0.09	6.87±0.05	8.82±0.18	6.84±0.13
TSIER	**23.91**±0.13	**3.62**±0.07	25.75±0.16	**34.31**±0.19	**27.18**±0.22	**7.20**±0.09	**9.69**±0.10	**7.88**±0.06
	TripAdvisor							
Att2Seq	14.90 ±0.09	0.93±0.11	14.22 ±0.15	18.69 ±0.21	15.08 ±0.13	2.01 ±0.03	2.53 ±0.05	2.05 ±0.08
ACMLM	8.13 ±0.13	0.21 ±0.02	11.63 ±0.16	12.70 ±0.04	9.91 ±0.10	0.85 ±0.04	0.72 ±0.06	0.65 ±0.09
NETE	17.40 ±0.21	2.12 ±0.05	20.55 ±0.13	31.10 ±0.14	23.56 ±0.27	4.93 ±0.08	7.78 ±0.13	5.48 ±0.18
PETER+	19.89 ±0.29	2.90 ±0.13	23.49 ±0.09	32.95 ±0.23	**26.02** ±0.12	6.05 ±0.15	8.13 ±0.08	6.85 ±0.14
T-GRU	20.12 ±0.17	2.87 ±0.10	23.02 ±0.25	32.11 ±0.18	25.03 ±0.31	6.12 ±0.10	7.94 ±0.16	6.78 ±0.12
TSIER	**21.53** ±0.10	**2.93** ±0.06	**23.57** ±0.19	**33.02** ±0.16	25.88 ±0.13	**6.31** ±0.12	**8.25** ±0.15	**7.10** ±0.08
	Amazon Movie							
Att2Seq	12.62 ±0.17	1.01 ±0.05	13.40 ±0.13	20.62 ±0.16	14.88 ±0.06	1.84 ±0.05	2.62 ±0.11	0.90 ±0.08
ACMLM	5.34 ±0.08	0.14 ±0.12	6.11 ±0.13	7.03 ±0.17	6.44 ±0.11	0.23 ±0.05	0.16 ±0.04	0.15 ±0.07
NETE	19.08 ±0.23	2.44 ±0.11	23.80 ±0.29	33.02 ±0.23	24.85 ±0.17	7.04 ±0.09	9.09 ±0.15	7.55 ±0.13
PETER+	22.58 ±0.08	3.72 ±0.05	24.91 ±0.16	35.45 ±0.08	26.48 ±0.12	8.40 ±0.17	10.23±0.19	7.72±0.14
T-GRU	22.17 ±0.12	3.41 ±0.07	23.38 ±0.11	34.80 ±0.13	24.92 ±0.08	6.61 ±0.10	7.51 ±0.12	6.91 ±0.10
TSIER	**22.8** ±0.20	**3.83** ±0.04	**25.14** ±0.13	**35.79** ±0.31	**26.85** ±0.14	**8.51** ±0.06	**10.42** ±0.08	**8.17** ±0.16

r,p,f represent Recall, Precision, and F1.
The best-performing values are presented in bold, and the second-best values are underlined.

mentioned above. TSIER is mainly divided into two parts: the filter-enhanced Transformer encoder layer responsible for sequential recommendation and the modified Transformer layer responsible for explanation generation. In this section, we calculate the parameter sizes of these two parts separately, and their sum.

We first calculate the parameter size of the Transformer with L layers. According to formulas (15) and (10), we have the weights for Q, K, and V ($W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$) and the weights for the multi-head attention ($W \in \mathbb{R}^{d\times d}$). Thus, the parameter size of the multi-head attention is $3\times d\times d + d\times d = 4d^2$. Each Transformer encoder block contains an FFN layer (formula (11)) and two layer normalization layers. For the FFN layer, we have $W_1 \in \mathbb{R}^{d\times 4d}$, $W_2 \in \mathbb{R}^{4d\times d}$, $b_1 \in \mathbb{R}^{4\times d}$, and $b_2 \in \mathbb{R}^{1\times d}$. The layer normalization contains $2\times 2\times d$ parameters. Thus, the parameter size of the FFN and layer normalization is $8\times d\times d+5\times d+4\times d = 8d^2+9d$. The parameter size of one Transformer encoder layer is $12d^2+9d$. Likewise, the parameter size of a Transformer decoder layer is $12d^2 + 9d + 4d^2 + 2d = 16d^2 + 11d$. Therefore, the parameter size of the Transformer with L layers is $L\times(12d^2+16d^2+9d+11d) = L(28d^2+20d)$.

- For the modified Transformer layers, the parameter size of input/output embedding is $|S|\times d$, and the parameter size of the last projection layer is $|V|\times d$. Thus, the parameter size of the modified Transformer layers is $L(28d^2+20d)+|S|d+|V|d \approx 37.45$ *million*.
- For the filter-enhanced Transformer encoder layer, the parameter size of input item embedding is $|\mathcal{I}|\times d$, where $\mathcal{I}$ represents the set of items. According to the Yelp dataset, we set $|\mathcal{I}| = 20,266$. The parameter size of the filter layer is $n\times d$. Thus, the parameter size of the filter-enhanced Transformer encoder layer is $L_f(12d^2+9d)+|\mathcal{I}|d+nd \approx 19.85$ *million*.
- According to the above calculations, the total parameter size of the proposed model TSIER is approximately $37.45M + 19.85M \approx 57.30$ *million*.

In conclusion, the total parameter size of our proposed model is approximately 57.3 million, with the explanation generation component alone accounting for about 37.45 million parameters. This is significantly lower than the parameter sizes of large pretrained language models, such as 117 million for GPT-1 and 110 million for BERT. This demonstrates the cost-effectiveness of our model as an unpretrained explanation-generating method.

VI. RESULTS AND DISCUSSION

A. OVERALL PERFORMANCE COMPARISON FOR THE GENERATED EXPLANATION (RQ1)

Table 3 and Table 4 compare the performances of the explanations generated by the baselines and our proposed method. The comparison shows that the performance of the ACMLM is low for text quality, except for USR. This result was obtained because the ACMLM is a model based on

BERT that learns from special mask mechanisms, which is different from the other baselines and our proposed method. The ACMLM generates an explanation by predicting masked tokens, which can further vary the generated sentences. However, this prediction mechanism is not useful when a generated sentence cannot be read correctly.

In terms of explainability, the ACMLM performs better in some cases because it can learn more from its mask mechanism, whereas the other models cannot attend to future tokens because the left-to-right mask prevents it. We also found that the performance of PETER+ on each dataset was the best or second best in a vast majority of the cases, outperforming NETE and Att2Seq in most cases. PETER+ generates explanations by establishing a relationship between the user ID and item ID. In addition, the mechanism of PETER+ for generating explanations follows a word-by-word approach, as in our proposed model, leading to performance similar to that of the proposed model in terms of text quality. Unlike PETER+, the proposed approach associates time-series information with recommended items and uses this information as a basis for generating explanations. This situation results in more features being enriched in the generated explanatory sentences because of the use of time-series information as the features of the input. Thus, our proposed model performs better than PETER+ in most cases. However, when our proposed method does not use time-series information or uses very little time-series information as the input, the generated explanation may not be as good as the sentences generated by PETER+. For example, it contains insufficient or incorrect features, which can lead to poor performance than that of PETER+ in terms of explainability.

In addition, we compared the proposed method with the GRU-based baselines NETE and T-GRU to demonstrate the competitiveness of our proposed method. The comparison results indicate that the performances of the GRU-based approaches are generally inferior to those of the Transformer-based approaches. The T-GRU performs well in terms of the DIV in some cases, probably because GRU is not sensitive to the mask mechanism compared with the Transformer model. Thus, it can learn more features during training.

According to the evaluation results of text quality and explainability, our proposed TSIER model performed well on all three experimental datasets, and it was considered effective in the explanation generation.

B. QUANTITATIVE ANALYSIS OF THE TIME-SERIES INFORMATION FEATURE MATCHING RATIO (RQ2)

For time-series information, we adopted the time-series information feature matching ratio (TFMR@K) for evaluation, which is the ratio of the time-series information from generated explanations that match the given time-series information features. K represents the latest K item in the purchase history sequence of a user. In this study, we set the value of K to 3 and 5 because all the datasets used in the experiments primarily consisted of short reviews. Typically, these reviews do not contain more than five time-series features per review. Table 5 lists the results of TFMR@3 and TFMR@5 on the three datasets.

The results confirm that TSIER performs better in most cases. It achieved the best performance on the Amazon Movie dataset because it had the largest average sequence length, which allowed the Transformer to learn time-series information effectively using the modified mask mechanism, helping the proposed approach outperform the GRU-based T-GRU. The performance of the other baseline models is lower because they do not consider time-series information. Therefore, the proposed approach performs well on the evaluation metric of time-series information if the sequence length is suitable.

TABLE 4. Results of explainability on three datasets. The experiments were conducted five times, with the mean and standard deviation reported.

	Explainability			
	FMR	FCR	DIV ↓	USR
	Yelp			
Att2Seq	0.05 ±0.02	0.12 ±0.03	2.27 ±0.06	0.11±0.03
ACMLM	0.13 ±0.04	0.42 ±0.02	0.93 ±0.02	**0.98**±0.03
NETE	0.83 ±0.01	0.36 ±0.04	0.95 ±0.04	0.74±0.05
PETER+	0.87 ±0.05	0.39 ±0.03	1.06 ±0.03	0.60±0.02
T-GRU	0.86 ±0.02	0.38 ±0.04	0.92 ±0.02	0.91±0.04
TSIER	**0.91**±0.02	**0.43**±0.03	**0.85**±0.05	0.94±0.02
	TripAdvisor			
Att2Seq	0.04 ±0.02	0.16 ±0.03	3.96 ±0.36	0.25 ±0.07
ACMLM	0.10 ±0.02	0.34 ±0.02	1.80 ±0.04	**0.89** ±0.03
NETE	0.63 ±0.04	0.14 ±0.05	0.96 ±0.08	0.60±0.03
PETER+	0.71 ±0.05	0.34 ±0.03	0.98 ±0.03	0.66±0.05
T-GRU	0.72 ±0.04	0.36 ±0.03	0.92 ±0.03	0.77 ±0.06
TSIER	**0.75** ±0.03	**0.37**±0.03	**0.91** ±0.02	0.85±0.05
	Amazon Movie			
Att2Seq	0.11 ±0.03	0.22 ±0.02	2.92 ±0.09	0.33 ±0.04
ACMLM	0.07 ±0.02	0.32 ±0.02	**0.90** ±0.04	**0.94** ±0.04
NETE	0.71 ±0.02	0.25 ±0.03	0.95 ±0.06	0.43 ±0.03
PETER+	0.82 ±0.05	0.28 ±0.02	1.10 ±0.02	0.68 ±0.04
T-GRU	0.82 ±0.01	0.41 ±0.02	1.02 ±0.04	0.72 ±0.02
TSIER	**0.83**±0.02	**0.51**±0.04	0.93±0.06	0.84±0.05

"↓" indicates that the lower the value, the better the performance.
The best-performing values are presented in bold, and the second-best values are underlined.

TABLE 5. Evaluation for our model and other explanation generation baselines on the proposed evaluation metric TFMR@K.

	Yelp		TripAdvisor		Amazon Movie	
TFMR@K	K=3	K=5	K=3	K=5	K=3	K=5
Att2Seq	0.00	0.00	0.01	0.00	0.00	0.00
ACMLM	0.02	0.00	0.03	0.00	0.05	0.01
NETE	0.07	0.01	0.04	0.00	0.08	0.01
PETER+	0.08	0.02	0.09	0.01	0.06	0.01
T-GRU	**0.41**	0.18	0.39	**0.08**	0.47	0.20
TSIER	**0.41**	**0.20**	**0.45**	0.07	**0.58**	**0.23**

The best-performing values in each row are presented in bold.

TABLE 6. Comparison of performance between the filter-enhanced Transformer encoder layer from our proposed model TSIER and other sequential recommendation baselines across three datasets. The experiments were conducted five times, and the mean and standard deviation were reported.

Datasets	Metric	BPR-MF	SASRec	SSE-PT	BERT4Rec	GRU4Rec	**TSIER**
Yelp	HR@5	0.089 ±0.012	0.583 ±0.015	0.561 ±0.021	0.616 ±0.014	0.501 ±0.024	**0.621** ±0.009
	HR@10	0.373 ±0.015	0.709 ±0.013	0.775 ±0.017	0.785 ±0.011	0.677 ±0.015	**0.808** ±0.011
	NDCG@5	0.161 ±0.005	0.394 ±0.006	0.403 ±0.008	0.423 ±0.005	0.342 ±0.004	**0.437** ±0.006
	NDCG@10	0.209 ±0.007	0.448 ±0.009	0.462 ±0.004	0.471 ±0.003	0.401 ±0.003	**0.475** ±0.003
TripAdvisor	HR@5	0.053±0.018	0.158 ±0.011	0.149 ±0.010	0.154 ±0.019	0.115 ±0.020	**0.166** ±0.015
	HR@10	0.144±0.009	0.274 ±0.007	0.353 ±0.007	0.355 ±0.013	0.290 ±0.013	**0.357** ±0.008
	NDCG@5	0.056 ±0.006	0.123 ±0.002	**0.144**±0.003	0.137 ±0.006	0.106 ±0.004	0.143 ±0.003
	NDCG@10	0.104 ±0.004	0.175 ±0.005	0.180 ±0.004	0.186 ±0.004	0.158 ±0.003	**0.194** ±0.005
Amazon Movie	HR@5	0.068 ±0.008	0.204 ±0.017	**0.253** ±0.013	0.234 ±0.019	0.190 ±0.005	0.239±0.021
	HR@10	0.144 ±0.005	0.279 ±0.021	0.328 ±0.005	0.326 ±0.008	0.245 ±0.011	**0.380** ±0.014
	NDCG@5	0.052 ±0.003	0.153 ±0.007	0.151 ±0.006	0.152 ±0.004	0.123 ±0.007	**0.169** ±0.006
	NDCG@10	0.096 ±0.005	0.181 ±0.003	0.189 ±0.004	0.180 ±0.006	0.148 ±0.002	**0.205**±0.007

The best-performing values are presented in bold, and the second-best values are underlined.

C. OVERALL PERFORMANCE COMPARISON FOR SEQUENTIAL RECOMMENDATION MODELS (RQ3)

The results of the proposed model and sequential recommendation baselines are shown in Table 6. In this study, we only applied N = 5 and 10 for NDCG@N and HR@N. The dimension of the hidden units was set to 512. The trend of the dimension change is discussed below.

Firstly, non-sequential recommendation methods such as BPR-MF exhibit lower performance compared to sequential recommendation methods, highlighting the importance of sequential patterns in this task. Among the sequential recommendation methods, Transformer-based architectures such as SASRec, BERT4Rec, and SSE-PT generally outperform RNN-based models such as GRU4Rec. This feature could be attributed to the larger number of parameters in Transformer-based models, enabling them to capture sequential characteristics more effectively. Additionally, SSE-PT shows comparable performance to SASRec and BERT4Rec on some datasets. This situation may be due to its use of stochastic shared embeddings, which help prevent overfitting, a challenge not adequately addressed by the existing regularization techniques such as layer normalization, dropout, and weight decay.

Our approach, TSIER, consistently outperforms all these baselines by a significant margin on most datasets. Unlike the baselines, TSIER uses a frequency domain architecture with learnable filters for encoding item sequences. These learnable filters mitigate the impacts of noise and function as circular convolutions, enabling the capture of periodic characteristics in item sequences with an expanded receptive field. Consequently, our proposed model surpasses the baselines in terms of both effectiveness and efficiency.

D. ABLATION STUDY (RQ4)

We analyzed all tasks in our approach via an ablation study to address RQ4. The results are listed in Table 7. We disabled each layer in our proposal separately on the Yelp dataset. All models in the ablation study used the same hyperparameters ($L = 3$ and $d = 512$). We introduce these variant models and analyze their effects. Further, to observe the differences between these ablation models and the original model more clearly, we have included results generated by some ablation models in the section VI-E (Case Study) for comparison.

TABLE 7. Ablation study on Yelp dataset, compared with default (TSIER), FL represents the filter layer, TSI and CP represent the time-series information and context prediction task, respectively.

Metrics	Default	w/o FL	w/o M	w/o CP	w/o TSI
FMR	0.89	0.80↓	0.75↓	0.84↓	0.62↓
FCR	0.44	0.32↓	0.14↓	0.30↓	0.17↓
DIV	0.83	0.90↓	3.61↓	1.98↓	2.52↓
USR	0.92	0.87↓	0.04↓	0.71↓	0.80↓
BLEU-1	23.79	23.74↓	20.31↓	22.40↓	24.03↑
BLEU-4	3.58	3.51↓	2.12↓	2.36↓	3.89↑
TFMR@3	0.41	0.33↓	0.34↓	0.37↓	0.14↓
TFMR@5	0.20	0.12↓	0.13↓	0.12↓	0.05↓
HR@10	0.805	0.713↓	0.811↑	0.809↑	0.002↓
NDCG@10	0.477	0.439↓	0.479↑	0.482↑	0.001↓

"↑" and "↓" represent increase and decrease in performance.

- Remove filter layer (w/o FL): We eliminated the filter layer and instead used a standard Transformer encoder layer to learn the representation of the input sequence. Compared to the baseline performance, the absence of frequency-domain transfer processing led to a significant drop in performance. This outcome verifies the necessity of the filter layer in enhancing the effectiveness of TSIER.
- Remove M (w/o M): Removing the modified attention mask M to the left-to-right masking resulted in a significant decrease in performance. This finding indicates that M plays a positive role in capturing time-series features. However, no noticeable change occurred in HR and NDCG, which could be due to the removal of M not significantly affecting the outcomes of the previous layers.
- Remove context prediction(w/o CP): After setting L_c to 0 at the multi-task learning stage, the performance

TABLE 8. One case from our proposed method, ablation models, and PETER+ on the Yelp dataset, which includes context words and explanations. All explanations are generated based on items predicted by the Transformer encoder layer with the filter layer from our proposal.

Ground Truth	
Item sequence	..., yakitori, fried eggplant, chicken pizza, **pork sandwich**;
Review	The environment is not loud and very comfortable, the meat is quite good and **pork sandwich** is sluggish, chicken is juicy.
Features	environment, meat, **pork**, sluggish, chicken
Item prediction results	
1. pork sandwich; 2. meatball pizza; 3. tonkatsu; 4. toyama sushi; 5.beef burger; ...	
Explanation generation results	
PETER+	The meat is good, and the chicken is juicy.
TSIER -w/o Filter layer	The **pork sandwich** is so good, friendly to English speakers.
TSIER -w/o TSI	The meat is good, environment is very comfortable, and I love this place.
TSIER	The **pork sandwich** is sluggish, **chicken** is quite good, I also enjoy the environment.
Context prediction results	
PETER+	⟨*eos*⟩, the, is, a, and,meat, so, it,I, chicken, loud, good, of, had, we
TSIER -w/o Filter layer	⟨*eos*⟩, the, a, and, is, **pork**, so, friendly, English, that, good, with, meat, very, in
TSIER -w/o TSI	⟨*eos*⟩, the, a, and, is, good, environment, quite, you, to, it, very, was, I, meat
TSIER	⟨*eos*⟩, the, a, **sandwich**, **chicken**, quite, with, **pork**, and, is, in, to, I ,sluggish, with

Features from the ground truth review and time-series information are underlined and boldfaced, respectively.

of all evaluation metrics decreased. This evidence demonstrated the essential role of the context prediction task in enhancing the readability and text richness of the generated explanations.

Since TSIER primarily predicts items and generates recommendation reasons based on historical purchase sequences, in this experiment, we mainly test what kind of recommendation reasons the model generates for an item without any historical purchase records (w/o TSI):

- We removed the time-series information by modifying the input of the Transformer, retaining only the last item as the recommended item to generate an explanation. Additionally, we padded the remaining items in the historical interaction sequence with '⟨*pad*⟩' and set L_r to 0 when generating explanations. This removal led to generated explanations that lacked time-series features, consequently reducing the model's effectiveness in predicting items compared to other sequential recommendation models. This absence explains the significant decrease in the performance metrics TFMR, HR, and NDCG. For increased BLEU metrics, it is not meaningful when the generated explanation is incorrect.

E. CASE STUDY (RQ4)

We also provide some cases generated in this study to address RQ4, as shown in Table 8. Initially, we input the historical purchase sequence of a user from the test set into the Transformer with a filter layer in our proposal, and it recommended a ''pork sandwich'', which matches the ground truth correctly. Subsequently, we allowed part of the ablation models and PETER+ to generate explanations for this recommendation. Additionally, we set this item as a target for TISER -w/o TSI to generate explanations. The review section presents the ground truth of the explanations generated below. The results indicate that each method successfully generated context words containing features from the ground truth. Notably, our proposed model, TISER, included time-series information like ''pork'' and ''chicken'', derived from the user's purchase history sequence. This suggests that the proposed method can effectively extract time-series features from the input sequence. The explanations are based on the predicted context words.

The explanations generated by PETER+ contain features from the ground truth but lack time-series information. For ablation models, when historical purchase behavior does not include the item (time-series information excluded), our model only generates explanations based on the recommended item and its features. This results in explanations that are devoid of time-series elements, as observed in the results from TSIER-w/o TSI. Additionally, when we remove the filter layer from our model (as seen in the results of TSIER-w/o Filter layer), the explanations still include time-series information. However, these explanations may not be suitable for the recommendation results as they can contain irrelevant details, such as ''friendly to English speakers''. In contrast, explanations generated by our proposed method, TSIER, effectively combine both ground truth features like ''environment'' and time-series information features such as ''pork sandwich''.Therefore, our approach successfully provides explanations that incorporate both time-series information and recommended items, enriching them with meaningful linguistic context.

F. HYPERPARAMETER SENSITIVITY (RQ5)

1) DIMENSIONS OF THE HIDDEN UNITS

We discuss how the dimensions of the hidden units affect the results. Given the consistent performance across different metrics, in Figure 5 we only show NDCG@10 and

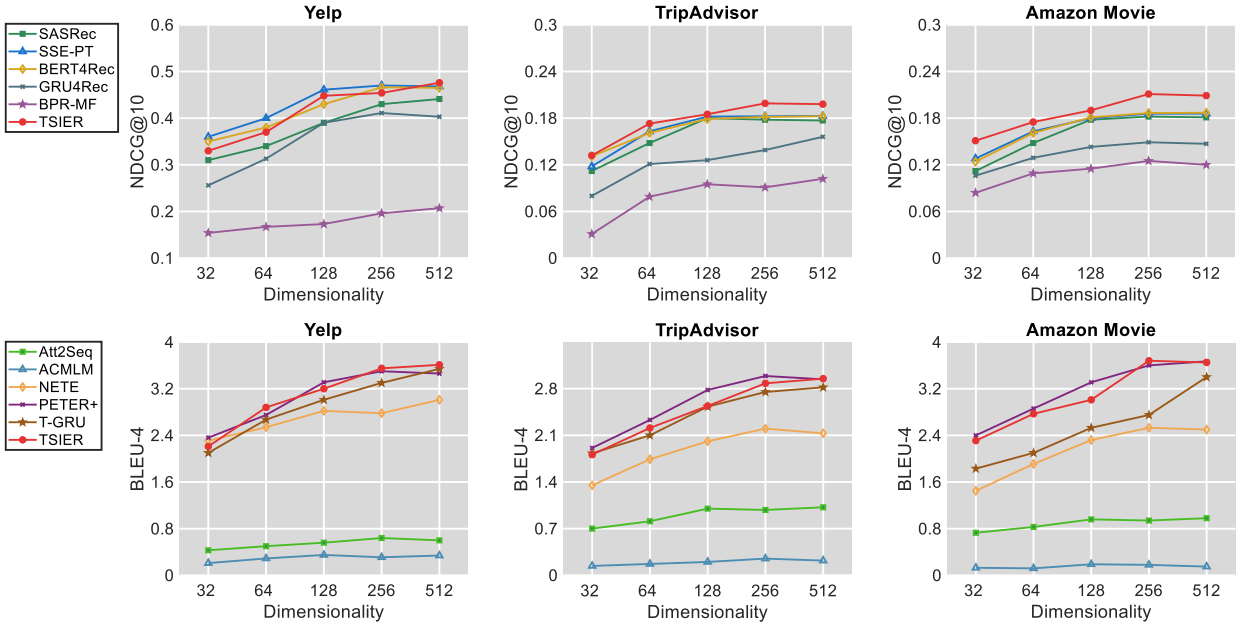

FIGURE 5. Results of the baseline models and proposed model on NDCG@10 and BLEU-4 across different dimensionalities of the hidden units.

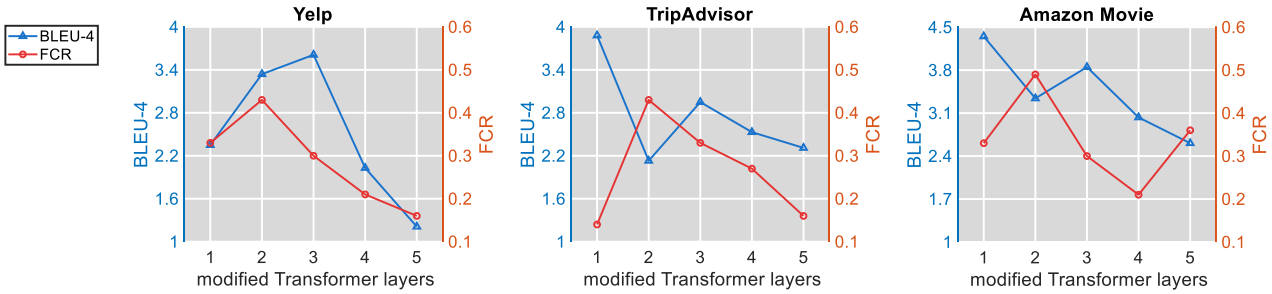

FIGURE 6. Results of the proposed model on BLEU-4 and FCR across different numbers of modified Transformer layers.

BLEU-4 with varied dimensions of the hidden units from {32, 64, 128, 256, 512} on three datasets, keeping the other hyperparameters constant. For NDCG@10, an improvement in model performance is noted with an increase in the hidden unit dimensions. However, when the dimension reaches 512, some models experience a performance decline, whereas others grow slowly, suggesting that dimensions larger than 256 may not be necessary for better performance. This feature is especially clear in the TripAdvisor dataset, which has the shortest average sequence length and may suggest overfitting. In terms of BLEU-4, both the Att2Seq and ACMLM models exhibit insensitivity to changes in the hidden unit dimensions. The performance trends of the other models align with those observed for NDCG@10. Although the performance of some models declined or grew slowly when dimensions were set to 512, to showcase the performance of our proposed model the most effectively, we opted for a dimension setting of 512.

2) NUMBER OF MODIFIED TRANSFORMER LAYERS

In this part of our proposed model, we focused on optimizing the neural network architecture by experimenting with various configurations of the modified Transformer layer. Our goal was to determine if increasing the number of layers would enable the network to learn more complex representations from the data. We tested a range of layers, including {1, 2, 3, 4, 5}. As shown in Figure 6, despite the goal of enhancing the complexity through additional layers, the BLEU-4 score for the modified Transformer layers peaks at the third layer, whereas the FCR reaches its maximum at the second layer for both the Yelp and Amazon datasets. Furthermore, for the TripAdvisor and Amazon datasets, the BLEU-4 score is the highest at the first layer. However, upon examining the explanations generated by our model with a single layer, it becomes apparent that these explanations contain repeated words and lack readability. A delicate balance must be struck: whereas a network with fewer layers might not capture enough complexity, a network that is too deep can lead to excessive computational costs and an increased risk of overfitting and may not always improve performance.

3) REGULARIZATION COEFFICIENT ω

In this study, we introduced two tasks to generate explanations based on time-series information. We examined how

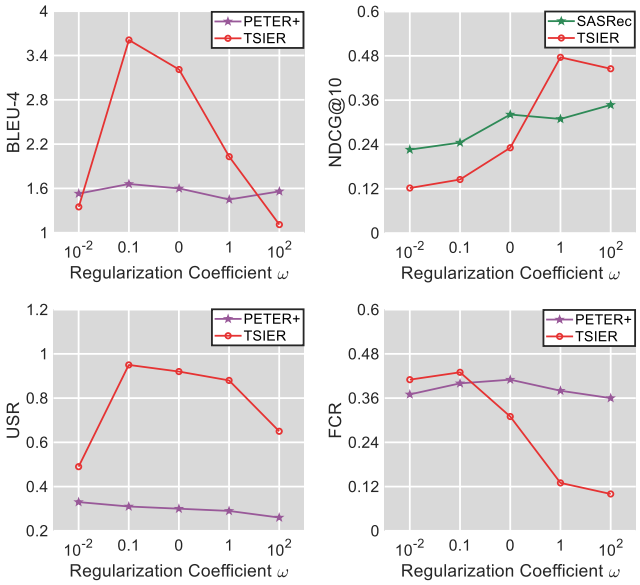

FIGURE 7. Impact of ω on the recommendation performance and explanation generation task and using TSIER on the Yelp dataset. To facilitate comparison, the results for PETER and SASRec are presented separately.

our proposed model responded to varying the regularization coefficient ω for the explanation generation and context prediction tasks, testing values from $\{10^{-2}, 0.1, 1, 10, 10^2\}$. For comparison, we included PETER+, noted for its SOTA performance and similar structure in explanation generation. To assess the text quality, we used BLEU-4 and the USR, and for explainability accuracy, we employed the FCR. Additionally, we monitored changes in NDCG@10 to evaluate the impact of ω on recommendation performance, comparing it with SASRec, which is also a Transformer-based model. In Figure 7, PETER+ shows minimal sensitivity to changes in the regularization coefficient ω, and the performance of SASRec is similar to that of our proposal. These findings indicate that a lower ω leads to a higher FCR and USR, a trend that is also evident in the BLEU-4 scores. However, at lower values of ω, such as 10^{-1}, although the performance of explanation generation peaks, the recommendation performance significantly suffers. This characteristic could be due to the model encountering a local minimum during training for sequence recommendation, preventing full optimization when ω is set too low. Consequently, in the tuning of ω for our proposed model, we prioritized text quality and explainability over the other metrics.

VII. CONCLUSION

In this study, we aimed to capture denoised time-series information from Transformer encoder layers and to use it to generate explanations that enhance both the expressiveness and the quality of recommendation explanations. To achieve this objective, we proposed a multitask Transformer-based model, TSIER, designed to generate explanations leveraging time-series information processed through filter-enhanced Transformer encoder layers. We conducted extensive experiments on three datasets, and the results demonstrated that our proposed method surpassed strong baselines by generating more effective and reasonable explanations for recommendation outcomes. Furthermore, we proposed an evaluation metric, TFMR@K, to assess whether the generated sentences contained time-series information. Additional experiments and analyses were performed, and the results confirmed that our approach could generate high-quality explanations enriched with time-series information features, thereby validating the effectiveness of our proposed method.

In future work, we intend to expand our approach to include multiple features for lengthy reviews, which often contain numerous instances of time-series information. The datasets employed in our experiments featured only brief reviews, lacking comprehensive time-series data. Consequently, the full potential of our proposed model, which thrives on abundant time-series information, may not have been completely showcased. We believe that this research could contribute to producing more varied explanations for users, enhancing their trust and satisfaction.

COMPETING INTERESTS

The authors have no financial or non-financial interests to disclose that are related to the content of this article.

AVAILABILITY OF DATA AND MATERIALS

The datasets and sentiment analysis method used in the experiments described in this paper can be accessed from https://github.com/qyp9909/TSIER-datasets.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
[2] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2014, pp. 83–92.
[3] J. Tan, S. Xu, Y. Ge, Y. Li, X. Chen, and Y. Zhang, "Counterfactual explainable recommendation," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 1784–1793.
[4] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam, "Neural rating regression with abstractive tips generation for recommendation," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 345–354.
[5] L. Li, Y. Zhang, and L. Chen, "Generate neural template explanations for recommendation," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 755–764.
[6] L. Li, Y. Zhang, and L. Chen, "Personalized transformer for explainable recommendation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics*, 2021, pp. 4947–4957.
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–16.
[8] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.
[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.
[10] W. Ahmad, S. Chakraborty, B. Ray, and K.-W. Chang, "Unified pre-training for program understanding and generation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 1–11.

[11] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NIPS*, 2020, pp. 1877–1901.

[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[13] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 188–197.

[14] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, Oct. 2009.

[15] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3926–3932.

[16] H. Zhao and X. Wang, "Bi-group Bayesian personalized ranking from implicit feedback," in *Proc. 2nd Int. Conf. Comput. Sci. Softw. Eng.*, May 2019, pp. 452–461.

[17] K. Zhou, H. Yu, W. X. Zhao, and J.-R. Wen, "Filter-enhanced MLP is all you need for sequential recommendation," in *Proc. ACM Web Conf.*, New York, NY, USA, Apr. 2022, pp. 2388–2399.

[18] Y. Zhang, H. Zhang, M. Zhang, Y. Liu, and S. Ma, "Do users rate or review? Boost phrase-level sentiment labeling with review-level sentiment classification," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2014, pp. 1027–1030.

[19] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *Found. Trends Inf. Retr.*, vol. 14, no. 1, pp. 1–101, 2020.

[20] N. Tintarev and J. Masthoff, "Explaining recommendations: Design and evaluation," in *Recommender Systems Handbook*. Cham, Switzerland: Springer, 2015, pp. 353–382.

[21] L. Chen, D. Yan, and F. Wang, "User evaluations on sentiment-based recommendation explanations," *ACM Trans. Interact. Intell. Syst.*, vol. 9, no. 4, pp. 1–38, Dec. 2019.

[22] F. Gedikli, D. Jannach, and M. Ge, "How should I explain? A comparison of different explanation types for recommender systems," *Int. J. Hum.-Comput. Stud.*, vol. 72, no. 4, pp. 367–382, Apr. 2014.

[23] X. Chen, Y. Zhang, and Z. Qin, "Dynamic explainable recommendation based on neural attentive models," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 53–60.

[24] L. Li, Y. Zhang, and L. Chen, "EXTRA: Explanation ranking datasets for explainable recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2021, pp. 2463–2469.

[25] L. Li, L. Chen, and R. Dong, "CAESAR: Context-aware explanation based on supervised attention for service recommendations," *J. Intell. Inf. Syst.*, vol. 57, no. 1, pp. 147–170, Aug. 2021.

[26] S. Shi, H. Chen, W. Ma, J. Mao, M. Zhang, and Y. Zhang, "Neural logic reasoning," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manag.*, 2020, pp. 1365–1374.

[27] H. Chen, S. Shi, Y. Li, and Y. Zhang, "Neural collaborative reasoning," in *Proc. Web Conf.*, Apr. 2021, pp. 1516–1527.

[28] X. He, T. Chen, M.-Y. Kan, and X. Chen, "TriRank: Review-aware explainable recommendation by modeling aspects," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 1661–1670.

[29] X. Wang, X. He, F. Feng, L. Nie, and T.-S. Chua, "TEM: Tree-enhanced embedding model for explainable recommendation," in *Proc. World Wide Web Conf. World Wide Web*, 2018, pp. 1543–1552.

[30] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," *Algorithms*, vol. 11, no. 9, p. 137, Sep. 2018.

[31] Y. Xian, Z. Fu, H. Zhao, Y. Ge, X. Chen, Q. Huang, S. Geng, Z. Qin, G. de Melo, S. Muthukrishnan, and Y. Zhang, "CAFE: Coarse-to-fine neural symbolic reasoning for explainable recommendation," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1645–1654.

[32] Z. Chen, X. Wang, X. Xie, T. Wu, G. Bu, Y. Wang, and E. Chen, "Co-attentive multi-task learning for explainable recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2137–2143.

[33] L. Dong, S. Huang, F. Wei, M. Lapata, M. Zhou, and K. Xu, "Learning to generate product reviews from attributes," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 623–632.

[34] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[35] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Apr. 2001, pp. 285–295.

[36] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 426–434.

[37] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[38] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[39] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016, pp. 1–17.

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[41] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 197–206.

[42] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1441–1450.

[43] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2001, pp. 311–318.

[44] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81.

[45] L. Wu, S. Li, C.-J. Hsieh, and J. Sharpnack, "SSE-PT: Sequential recommendation via personalized transformer," in *Proc. 14th ACM Conf. Recommender Syst.*, Sep. 2020, pp. 328–337.

[46] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 889–898.

YUANPENG QU received the M.S. degree from the University of Tsukuba, Japan, in 2023, where he is currently pursuing the Ph.D. degree in intelligent and mechanical interaction systems. His research interests include generative recommender systems and natural language generation.

HAJIME NOBUHARA (Member, IEEE) received the Dr.Eng. degree in computational intelligence from Tokyo Institute of Technology, Japan, in 2003.

From October 2002 to March 2006, he was an Assistant Professor with Tokyo Institute of Technology. In 2006, he became an Assistant Professor with the University of Tsukuba and established Computational Intelligence and Multimedia Laboratory. In 2013, he became an Associate Professor with the University of Tsukuba, where he has been a Professor, since 2022. His research interests include computational intelligence, image processing, web intelligence, bioinformatics, and UAV.