Transformer Reasoning Network for Personalized Review Summarization

Hongyan Xu
College of Intelligence and Computing, Tianjin University
Tianjin, China
hongyanxu@tju.edu.cn

Hongtao Liu*
College of Intelligence and Computing, Tianjin University
Tianjin, China
htliu@tju.edu.cn

Pengfei Jiao
Center for Biosafety Research and Strategy, Law School, Tianjin University
Tianjin, China
pjiao@tju.edu.cn

Wenjun Wang†
College of Intelligence and Computing, Tianjin University
Tianjin, China
College of Information Science and Technology, Shihezi University
Xinjiang, China
wjwang@tju.edu.cn

ABSTRACT

Review summarization aims to generate condensed text for online product reviews, and has attracted more and more attention in E-commerce platforms. In addition to the input review, the quality of generated summaries is highly related to the characteristics of users and products, e.g., their historical summaries, which could provide useful clues for the target summary generation. However, most previous works ignore the underlying interaction between the given input review and the corresponding historical summaries. Therefore, we aim to explore how to effectively incorporate the history information into the summary generation. In this paper, we propose a novel transformer-based reasoning framework for personalized review summarization. We design an elaborately adapted transformer network containing an encoder and a decoder, to fully infer the important and informative parts among the historical summaries in terms of the input review to generate more comprehensive summaries. In the encoder of our approach, we develop an inter- and intra-attention to involve the history information selectively to learn the personalized representation of the input review. In the decoder part, we propose to incorporate the constructed reasoning memory learning from historical summaries into the original transformer decoder, and design a memory-decoder attention module to retrieve more useful information for the final summary generation. Extensive experiments are conducted and the results show our approach could generate more reasonable summaries for recommendation, and outperform many competitive baseline methods.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Personalized Review Summarization; Reasoning Network; Transformer; Recommender system

ACM Reference Format:
Hongyan Xu, Hongtao Liu, Pengfei Jiao, and Wenjun Wang. 2021. Transformer Reasoning Network for Personalized Review Summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462854

1 INTRODUCTION

In recommender system, product review summaries E-commerce platforms (e.g., Amazon) are known as 'tips' or 'headline', and review summarization aims to generate a brief summary for the given online product review. This can help other users make reasonable purchase decisions quickly and alleviate the information overload problem [14]. Therefore, the review summarization task has received more and more attention recently in both industry and academia fields.

Different from the pure text summarization in natural language processing, review summarization in E-commerce platforms is highly challenging and personalized since there is other important recommendation information to be considered, such as the personalized characteristics of users and products besides the given input reviews. Recently, researchers have proposed many works to generate high-quality summaries for online reviews. Some methods utilize attributes of the given review, such as the ID and history information of the corresponding user/product to strength the personality of the generated summary. For example, Li et al. [14] propose to adopt user embedding to select user-preferred words in encoder and design a user-specific vocabulary to generate personalized summaries. And, some other methods leverage ratings given by users to control the sentiment tendency of the generated summaries. Ma et al. [25] and Chan et al. [3] both jointly optimize

*Hongtao Liu contributes equally with Hongyan Xu and is the co-first author
†Corresponding Author: Wenjun Wang

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462854

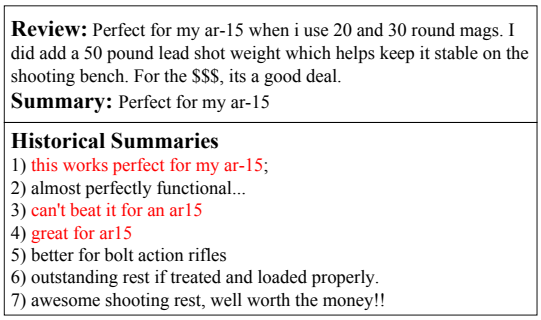

Figure 1: An example of product review and its corresponding summary and historical summaries of corresponding user and product. We mark the relevant historical summaries in red.

review summarization and sentiment classification tasks by treating rating score along with the given review as the sentiment label.

These methods have achieved great improvements in review summarization by utilizing these characteristics. However, existing approaches usually simply adopt the historical information as additional features along with the given input review to generate the target summary, and neglect the deep interaction between them yet. In fact, the history summaries of users and products have strong relatedness with the current summary since a user always have similar writing style in summaries. Therefore, fully capturing the complicated relevance between the given review and historical summaries is essential in personalized review summarization. Furthermore, different historical summaries are of different informativeness i.e., some of the historical summaries of the current user and product are less relevant or irrelevant with the input review as well as the summary generation. For example, as shown in Figure 1, we can see that some historical summaries are highly relevant with the current summary, e.g., they all mention "this product fits to ar-15". Besides, in Figure 2, there are nearly 30% target summaries words that appear in both input reviews and the corresponding historical summaries. Hence, it is necessary to reason and retrieve important information from these histories in terms of the given review rather than regarding them equally.

Besides, another important observation is that historical summaries are beneficial to alleviate the out-of-vocabulary problem in summarization task. As shown in Figure 2, historical summaries have nearly 25% common words with the target summary, while these words are not in input reviews. Especially, the historical summaries have many useful sentiment words (e.g., "nice", "recommend") and important aspect words (e.g., "quality", "price"). Therefore, it is helpful for review summarization by incorporating these words into the personalized dynamic vocabulary.

Based on the above motivations, in this paper, we propose a novel **T**ransformer **R**easoning **N**etwork for personalized review **S**ummarization (named TRNS). It is partially inspired by the powerful transformer network [29] which can capture the complex relevance among different elements and learn contextualized representations for them. In our approach, we design an adapted transformer architecture, which contains two reasoning units: 1) in the encoder, we conduct the reasoning attention module infer useful information from the historical summaries to learn more comprehensive representation for the given input review. 2) in the decoder, we first construct a personalized dynamic vocabulary from the historical summaries, and design another reasoning attention module to generate target words that not in the vocabulary.

Firstly, in the encoder, we first design an inter-reasoning self-attention among the historical summary documents of the current user and product, and the query of the attention is the feature of the input review; then the output is the history-aware representation for the input review. Through the inter-reasoning attention layer, our model can help reason more important parts or words in historical summaries in terms of the input review. Afterward, a personalized intra-reasoning attention is employed to select informative words in the input review, since different words in the review usually have different informativeness for generating the summary. Note that we utilize the personalized features (i.e., the IDs of user/product and ratings) as the query in the intra-reasoning attention; in this way, we can obtain more personalized weights for all words in the input review. The final representations of the input review could be obtained.

Secondly, in the decoder, we first construct a personalized dynamic vocabulary for each input review by filtering some high-frequency words from the historical reviews and summaries of the current user and product. To generate personalized words, we design a reasoning attention among the vocabulary in decoder layers and the query of the attention is the semantic representation at current decoding step. In this way, our model augments the original decoder in transformer network with the ability to predict words from the constructed vocabulary in terms of the given review.

Above all, through our proposed transformer reasoning network, we can not only effectively capture the underlying interaction information automatically between the history summaries and the input review, but make more exact reasoning to differently focus on important parts for a better target summary generation.

We summarize our main contributions as follows: (1) to the best our knowledge, we are the first to conduct review summarization by capturing the deep interaction between the given review and the corresponding historical summaries. (2) we propose a transformer-based method for review summarization which mainly contains two reasoning units which infer the useful information from the historical summaries to learn more comprehension review representation and predict words from the personalized dynamic vocabulary in terms of the given review respectively. (3) the experimental results on five benchmark datasets show that our model outperforms the state-of-the-art review summarization models.

2 RELATED WORK

2.1 Review Summarization

Review summarization is an important task in the recommender system and natural language processing, which aims to generate brief summary for the online product review. Different from the previous text summarization methods [9, 38], product reviews usually have various personalized information (e.g., rating, user and product information) which plays a crucial role in the text generation task in recommendation [6, 35].

Table 1: Characteristics of different models. Especially, "Historical text" denotes the historical reviews or summaries of user/product. And, "Interaction" denotes the relevance between the input review and the historical summaries.

	S2S+Attn [1]	PGN [27]	HSSC [25]	PATG [18]	memAttr [22]	USN [14]	Dual-View [3]	**TRNS**
User ID	×	×	×	×	√	√	×	√
Product ID	×	×	×	×	√	×	×	√
Rating	×	×	√	√	×	×	√	√
Source Review	√	√	√	×	√	√	√	√
Historical Text	×	×	×	√	√	√	×	√
Interaction	×	×	×	×	×	×	×	√

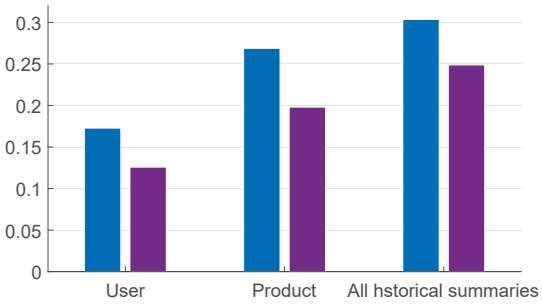

Figure 2: We count the proportion of two types words in the target summary. Especially, we list the results of the historical summaries from users and products respectively. The blue bars represent the target summary words that appear both in the given review and the historical summaries. The purple bars represent the target summary words that appear in historical summaries but not in the given review.

There have been many review summarization works [10, 12, 31] which mainly contains extractive and abstractive approaches. Some methods extract the important components from input review as summary [8, 34]. For example, Xiong et al. [34] propose an extractive method which exploits the review helpfulness information. However, some previous researches [2, 5] show that abstractive methods tend to be more effective than extractive methods for review text. Thus, we mainly focus on abstractive methods.

Recently, some abstractive approaches [3, 15, 22] are proposed for review summarization. Li et al. [19] and Li et al. [18] generate the summaries from the historical text of the current user and product, and conduct rating prediction to control the sentiment tendency in the generation process. In contrast to it, some methods treat the given review as the input and utilize the user/product information to enhance the summary generation. Li et al. [14] design a selective mechanism which utilizes user embedding to select user preference words, and generates summaries from the user-specific vocabulary memory. In addition, some methods also leverage aspect information to enhance review summarization [28, 35].

However, most of them ignore the deep fusion of the historical text and personalized information. Liu et al. [22] propose a memory network that utilizes the input review as the query, the historical review as key and the corresponding historical summaries as value to capture the user and product information, then feed the learned context vectors into the RNN-based decoder. Different from it, we take advantage of the interaction between the input review and the corresponding summaries to learn more comprehension review representation. In particular, we design reasoning units both in encoder and decoder, which can infer the important information from the historical summaries in terms of the input review and generate words from the personalized dynamic vocabulary constructed from the historical text. We list characteristics of several advanced methods and our model in Table 1.

2.2 Transformer Network

Different from LSTM [11] and GRU [4], Transformer [29] is based solely on the attention mechanism and achieves great performance with the ability to capture long-distance dependencies. Recently, Transformer has been widely used in text summarization [7, 16, 24], text generation [17], image caption [37], and other natural language processing tasks. You et al. [36] propose a focus-attention mechanism to learn document representation and an independent saliency-selection network to manage the information flow from encoder to decoder. However, these summarization approaches ignore the deeply interaction between the input review and the personalized information (e.g., historical summaries) along with it. In this paper, we augments Transformer to conduct review summarization, which derives history-aware representations from the historical summaries and selects the salient components from the input review by utilizing the personalized feature (i.e., the ID of user/product and rating) as the query.

3 PROPOSED METHOD

In this section, we first introduce the problem formulation and the original transformer framework. Then, we describe our proposed method from two aspects: 1) the personalized encoder layer with inter-reasoning attention to infer the important parts from the historical summaries, and intra-reasoning attention to infer the salient components for the input review; 2) the decoder with the historical reasoning memory which incorporates the historical text into the decoder to generate personalized summaries. Figure 3 illustrates the overall architecture of our model.

3.1 Problem Formulation

Give a review and the corresponding attributes (i.e., ID of user and product, rating and the historical information of users/products), our model aims to generate a summary $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{\hat{T}}\}$, where $\hat{T}$ is the length of the generated summary. The reference summary sequence is denoted as $Y = \{y_1, y_2, \cdots, y_T\}$, where T is the length of the reference summary. The input review is represented as $X =$

$\{w_1, w_2, \cdots, w_L\}$ where L is the length of the input review. The input review and the target summary share the vocabulary V which is constructed from the dataset. And the word embedding $\mathbf{w}_i$ of each word is extracted from the word embedding matrix $\mathbf{V} \in \mathcal{R}^{|V| \times d_w}$, where d_w denotes the dimension of word embeddings. User/product ID embeddings are widely used in recommender systems, which can indicate their intrinsic characteristics [23, 32, 33]. In this paper, the ID embedding of user u is denoted as $\mathbf{u} \in \mathcal{R}^{d_c}$, the ID embedding of product p is $\mathbf{p} \in \mathcal{R}^{d_c}$ where d_c denotes the dimension of the embedding. In addition, we convert the ratings, ranging in $[1, 5]$, into a low dimension dense vector denoted as $\mathbf{r} \in \mathcal{R}^{d_c}$, We further construct a historical summary set S_{uv} for each review by collecting K summaries of the corresponding user u and product v.

3.2 Transformer Network

Transformer is proposed in [29], which has achieved great success in various tasks. Transformer is an encoder-decoder framework which could well capture the deep interaction between words in a sentence. Hence, we design an adapted transformer model for review summary generation. In this section, we will present the transformer network briefly.

3.2.1 Multi-head Self-Attention Mechanism. Given input elements $\mathbf{G}$, self-attention mechanism aims to learn contextualized representations for given elements by modeling the dependency among them. Therefore, the query $\mathbf{Q}$, key $\mathbf{K}$ and value $\mathbf{V}$ are all the linear projections of $\mathbf{g}$. The output of self-attention is calculated following the scaled dot-product attention:

$$ATT(\mathbf{G}) = Softmax(\frac{(\mathbf{G}\mathbf{W}^Q)^T(\mathbf{G}\mathbf{W}^K)}{\sqrt{d_e}})(\mathbf{G}\mathbf{W}^V), \quad (1)$$

where d_e is the scaling factor and $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$ are learnable parameter matrices. Besides, the self-attention in Transformer adopts the h heads parallel implementation, where each head calculates the attention based on Equation (1). The output of the multi-head attention is the concatenation of h heads followed by a linear projection:

$$MultiHead(\mathbf{G}) = f(\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_h)\mathbf{W}^M \quad (2)$$

$$\mathbf{H}_i = ATT(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \quad (3)$$

where f is a concatenation operation. And $\mathbf{W}_i^Q \in \mathcal{R}^{\frac{d_e}{h} \times d_e}$, $\mathbf{W}_i^K \in \mathcal{R}^{\frac{d_e}{h} \times d_e}$, $\mathbf{W}_i^V \in \mathcal{R}^{\frac{d_e}{h} \times d_e}$ and $\mathbf{W}^M \in \mathcal{R}^{d_e \times d_e}$ are learnable parameter matrices.

3.2.2 Encoder-Decoder Architecture. Given the input review $X = \{w_1, w_2, \cdots, w_L\}$ where L is the length of the review, the encoder aims to learn contextualized representations $\mathbf{E}$ for all words in the review. The encoder is consists of a stack of N_1 identical layers. Each layer has two sub-layers: a self-attention mechanism and a fully connected feed-forward network [29]. Note that the input of the first layer is the embedding of the input review $\mathbf{E}_0 = \mathbf{X}$.

Based on the learned review representation $\mathbf{E}$ and the partial generated summaries, the decoder generates the summary $\hat{Y}$ word by word. The decoder is consists of N_2 identical layers. In each decoder layer, in addition to the two sub-layers in the encoder layer, there is also an encoder-decoder self-attention, which is between

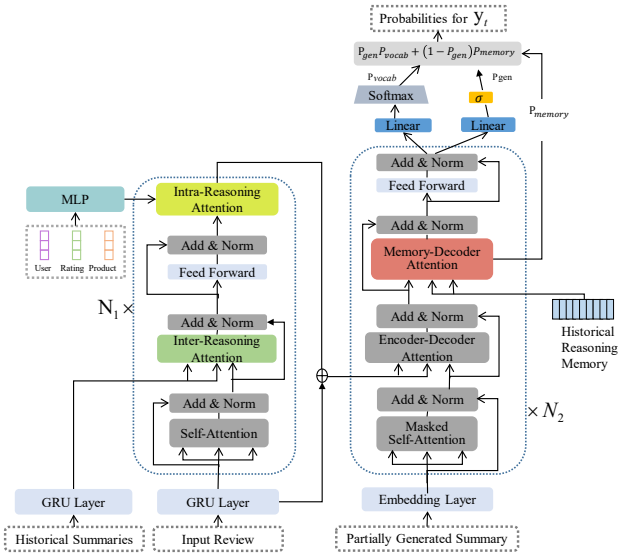

Figure 3: The framework of our model. Attentions marked in grey are from the naive Transformer. Then, the reasoning unit is consists of the inter-reasoning attention marked in green and the personalized intra-reasoning attention marked in yellow. Finally, the memory-decoder attention marked in red incorporates the historical reasoning memory into the decoder layer.

self-attention sub-layer and the feed-forward network. Next, we will present our well-designed reasoning attention units based on the Transformer.

3.3 Personalized Encoder Layer

In this paper, we augments the original Transformer elaborately to the personalized review summarization task with the ability to capture the interaction between the input review and the personalized information of it. As shown in the left of Figure 3, the encoder of our model is composed of N_1 identical layers. Given a review and corresponding personalized attributes (i.e., historical summaries, user id, product id and rating), the encoder aims to acquire history-aware representation for the review by capturing the complex relevance among them.

In this section, we would introduce our proposed personalized encoder layer. Each layer contains three sub-layers: 1) review self-attention sub-layer to learn representations for the input review; 2) inter-reasoning attention sub-layer to infer the useful information from the historical summaries; 3) intra-reasoning mechanism to infer the salient components of the input review by using the personalized feature (i.e., rating, the ID of user and product) as the query.

3.3.1 Review Encoding. This module aims to get the representations for the given review and the historical summaries of the corresponding user and product. The input review is first feed into a bi-directional GRU [4] layer to learn the semantic representation, and it could be represented as $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_L\}$, where

$\mathbf{h}_i \in \mathcal{R}^{1\times d_e}$, L is the length of the input review. For historical summaries, we first combine all the historical summaries in S_{uv} into a long sequence which is also feed into the GRU layer to learn the semantic representation and it can be denoted as $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{L'}\}$, where $\mathbf{z}_i \in \mathcal{R}^{1\times d_e}$, L' is the length of the sequence.

Inspired by [21], we further conduct self-attention over words in the input review to capture the long-distance dependence. It can be calculated as followings:

$$ATT(\mathbf{H}) = Softmax(\frac{(\mathbf{H}\mathbf{W}_r^Q)^T(\mathbf{H}\mathbf{W}_r^K)}{\sqrt{d_e}})(\mathbf{H}\mathbf{W}_r^V), \tag{4}$$

where, $\mathbf{W}_r^Q$, $\mathbf{W}_r^K$, and $\mathbf{W}_r^V$ are learnable parameters. Then, the given review is represented as $\mathbf{E}_{ATT}$.

3.3.2 Inter-Reasoning Attention. This module aims to learn a history-aware representation for the given review. In particular, we design an inter-reasoning attention module to infer the useful information from the historical summaries in terms of the given review. In the inter-reasoning attention, the query $\mathbf{Q}_{inter}$ is the linear projection of the review representation $\mathbf{E}_{ATT}$ from the last sub-layer. And, the key $\mathbf{K}_{inter}$ and value matrices $\mathbf{V}_{inter}$ are both the linear projection of historical summaries representations $\mathbf{Z}$. Then the attention is calculated as follows:

$$ATT(\mathbf{Q}_{inter}, \mathbf{K}_{inter}, \mathbf{V}_{inter}) = Softmax(\frac{\mathbf{Q}_{inter}^T\mathbf{K}_{inter}}{\sqrt{d_e}})\mathbf{V}_{inter}, \tag{5}$$

$$\mathbf{Q}_{inter} = \mathbf{E}_{ATT}\mathbf{W}_{e0}^l, \tag{6}$$

$$\mathbf{K}_{inter} = \mathbf{Z}\mathbf{W}_{e1}^l, \tag{7}$$

$$\mathbf{V}_{inter} = \mathbf{Z}\mathbf{W}_{e2}^l, \tag{8}$$

where $\mathbf{W}_{e0}^l$, $\mathbf{W}_{e1}^l$, $\mathbf{W}_{e2}^l$ are learnable parameters. In this way, our model is able to identify the useful information of the historical summaries relevant to the given review, e.g., semantically similar words. And, this module could get the history-aware review representation $\mathbf{E}_{inter}$. Afterwards, residual connection and layer normalization is employed:

$$\mathbf{E'}_{inter} = LayerNorm(\mathbf{E}_{ATT} + \mathbf{E}_{inter}). \tag{9}$$

3.3.3 Intra-Reasoning Attention. Furthermore, we employ a personalized intra-attention mechanism to infer the salient parts of the input review that should be focused on in the generation process. Specially, the query of this attention is the feature vector learned from the personalized information, including user ID embedding $\mathbf{u} \in \mathcal{R}^{d_c}$, product ID embedding $\mathbf{p} \in \mathcal{R}^{d_c}$ and rating embedding $\mathbf{r} \in \mathcal{R}^{d_c}$. It can be formalized as:

$$\mathbf{q} = MLP([\mathbf{u}; \mathbf{p}; \mathbf{r}]), \tag{10}$$

where $\mathbf{q} \in \mathcal{R}^{1\times d_e}$ is the feature vector that can be used in the reasoning process. In the personalized intra-attention, the key and value matrices are both the review representations from the inter-reasoning attention sub-layer $\mathbf{E'}_{inter}$. The output of this attention is calculated as follows:

$$\begin{aligned} \alpha_i &= f([\mathbf{E'}^i_{inter}; \mathbf{q}]), \\ \mathbf{E}^i_{intra} &= \mathbf{E'}^i_{inter} \odot \alpha_i, \end{aligned} \tag{11}$$

where f is sigmoid function and $\alpha_i \in \mathcal{R}^{1\times d_e}$ is the weight vector for the i-th words in the input review. After N_1 encoder layers, the output of the encoder is $\mathbf{E} \in \mathcal{R}^{L\times d_e}$, where L is the length of the given review.

To obtain more comprehension representation for the given review, we concatenate semantic representation directly from the input review $\mathbf{H}$ and the reasoned representation from the historical summaries $\mathbf{E}$:

$$\mathbf{E'} = f([\mathbf{E}; \mathbf{H}]), \tag{12}$$

where f is the linear function, and $\mathbf{E'}$ is the final output of the encoder for all words in the input review, which will be used to generate personalized summaries.

3.4 Decoder with Historical Reasoning Memory

Based on the learned review representation $\mathbf{E}$, the decoder generates a personalized summary word by word. The decoder is consists of N_2 identical layers. In the decoder, the input is partially generated summaries and the representations of words in the input are the sum of the word embedding and the position embedding [29].

To further enhance the summary generation, we design a historical reasoning memory and incorporate it into the decoder through the memory-decoder self-attention. In particular, we first construct a personalized dynamic vocabulary V_d for each review from the historical documents (i.e., historical reviews and summaries) of the corresponding user and product. Then, the historical reasoning memory $\mathbf{M}$ is extracted from word embedding matrices $\mathbf{V}$ with the word id in vocabulary V_d. The historical reasoning memory can be denoted as $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_{|V_d|}\}$, where $|V_d|$ is the memory size, $\mathbf{m}_i \in \mathcal{R}^{1\times d_w}$ is the word embedding for i-th word in the vocabulary V_d. In each decoder layer, we design a memory-decoder self-attention to conduct addressing and reading operation on the memory $\mathbf{M}$.

At time step t, the partial generated summary is encoded through the masked self-attention and then the second self-attention aims to align between the encoder and decoder. The decoder state is represented as $\mathbf{S}_t \in \mathcal{R}^{t\times d_e}$.

Afterwards, the memory-decoder attention is applied over the memory to select the important words relevant to current decoder state. Therefore, the query matrices $\mathbf{Q}_M$ is from the linear projection of the output of the encoder-decoder sub-layer $\mathbf{S}_t$. And, the key $\mathbf{K}_M$ and value matrices $\mathbf{V}_M$ are both from the linear projection of the embedding of the historical reasoning memory $\mathbf{M}$. Then, the output of this attention is calculated as follows:

$$ATT(\mathbf{S}_t, \mathbf{M}, \mathbf{M}) = Softmax(\frac{\mathbf{Q}_M^T\mathbf{K}_M}{\sqrt{d_e}})\mathbf{V}_M, \tag{13}$$

$$\mathbf{Q}_\mathbf{M} = \mathbf{S}_t\mathbf{W}_{m0}^l, \tag{14}$$

$$\mathbf{K}_\mathbf{M} = \mathbf{M}\mathbf{W}_{m1}^l, \tag{15}$$

$$\mathbf{V}_\mathbf{M} = \mathbf{M}\mathbf{W}_{m2}^l, \tag{16}$$

where $\mathbf{W}_{m0}^l \in \mathcal{R}^{d_e\times d_e}$, $\mathbf{W}_{m1}^l \in \mathcal{R}^{d_w\times d_e}$, $\mathbf{W}_{m2}^l \in \mathcal{R}^{d_w\times d_e}$ are learnable parameters and d_w is the dimension of the word embedding. After N_2 layers, the output of the decoder at time step t is $\mathbf{S}_t^{N_2}$.

To generate personalized words that are not in the input review, we introduce a pointer network [30] to copy the target words from the historical reasoning memory. Inspired by the Pointer Generator Network [27], there is also a probability P_{gen}, which decides to generate from the sharing vocabulary or copy from the constructed memory above:

$$P_{gen} = \sigma(\mathbf{S}_t^{N_2}\mathbf{W}_p + \mathbf{b}_p)\,, \tag{17}$$

where $\mathbf{W}_p \in \mathcal{R}^{d_e \times 1}$ and $\mathbf{b}_p \in \mathcal{R}^1$ are learnable parameter matrices. Then, the probability distribution P_{vocab} over the fixed vocabulary can be calculated like followings:

$$P_{vocab} = Softmax(\mathbf{W}_v \mathbf{S}_t^{N_2} + \mathbf{b}_v), \tag{18}$$

where $\mathbf{W}_v \in \mathcal{R}^{d_e \times |V|}$ and $\mathbf{b}_v \in \mathcal{R}^{1 \times |V|}$ are learnable parameter matrices. The final distribution for the target word prediction is the weighted sum of the distribution over the fixed vocabulary P_{vocab} and the distribution over the historical reasoning memory:

$$P(\hat{y}_t) = P_{gen}P_{vocab} + (1 - P_{gen}) \sum_{i:V_{d_i}=y_t} \beta_{ti}\,, \tag{19}$$

where β_{ti} is the attention weights from the memory-decoder attention (i.e., Equation (13)) from the last decoder layer. We use the negative log-likelihood as the loss function (NLLLoss) to train the review summary generation model:

$$\mathcal{L}_\phi(\hat{Y}|X) = \sum_{t=0}^{\hat{T}} -logP(\hat{y}_t)\,, \tag{20}$$

where $\hat{T}$ is the length of the generated review summary and ϕ is model parameters.

4 EXPERIMENTS

4.1 Datasets

To evaluate our method, we conduct extensive experiments on five popular datasets from Amazon[1]: **Toys and Games**, **Sports and Outdoors**, **Home and Kitchen**, **Electronics**, **Movies and TV**. Each case of these datasets contains the user ID, product ID, rating, review text, and summary text. In this paper, we only reserve the reviews given by active users to popular products, where each user and each product has at least K historical summaries. We discard reviews that have less than K historical summaries. For reviews that have more than K historical summaries, we select top-K historical summaries that have more common words with the input review. Furthermore, considering the different data sizes, we select different numbers of historical summaries: we set $K = 5$ for *Toys* and *Sports* dataset, $K = 10$ for *Home* and *Elect* dataset and $K = 20$ for *Movie* dataset. The length of each review is set to $L = 500$ and We keep the length of each summary $T = 15$. Following previous work [25], we randomly select 1000 samples for the validation set and test set separately, and the rest of the dataset as the training set. The dataset statistics are listed in Table 2.

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

Table 2: Dataset statistics.

Dataset	Users	Products	Reviews
Toys and Games	19,412	11,924	167,504
Sports and Outdoors	35,598	18,357	296,214
Home and Kitchen	66,212	27,991	550,461
Electronic	191,522	62,333	1,684,779
Movies and TV	123,960	50,052	1,697,471

4.2 Evaluation Metric

Following the previous works, we utilize the widely used metric ROUGE [20] as our evaluation metric to evaluate the quality of generated summaries. Following previous works [25, 27], we report F-measures of ROUGE-1, ROUGE-2, and ROUGE-L in our experiments. ROUGE-1 and ROUGE-2 counts the overlapping of uni-grams and bi-grams between the generated summaries $\hat{\mathbf{Y}}$ and the references $\mathbf{Y}$ given by users. ROUGE-L counts the longest common subsequence between the generated summaries $\hat{\mathbf{Y}}$ and the references $\mathbf{Y}$.

4.3 Baseline Methods

To evaluate the performance of our method, we compare it with several competitive review summarization methods:

- **TextRank**(Mihalcea et al. 2004): an extractive method that ranks sentences with the graph-based algorithm.
- **S2S-att** [1]: sequence-to-sequence model with the attention mechanism based on the bidirectional GRU network.
- **HSSC** [25]: a joint framework for abstractive summarization and sentiment classification.
- **PGN** [27]: a popular abstractive summarization method with a copy mechanism to generate words from the input reviews.
- **memAttr** [22]: a neural method that leverages the historical text of users and products to enhance the model performance.
- **USN** [14]: a personalized review summarization model, which considers the user's writing style and preference on different aspects of the product.
- **Dual-view** [3]: a very-recent dual-view model to jointly improve the review summarization and sentiment classification tasks and introduce an inconsistency loss in training to make the generated summary be consistent with the input review with aspect to the sentiment tendency.
- **Transformer** [29]: an encoder-decoder structure based solely on the attention mechanism and it generates summaries only based on the input reviews.

4.4 Implementation Details

We build vocabulary for each dataset separately by removing the stopping words and reserving the high-frequency words, meanwhile, the encoder and decoder module share the same vocabulary. Furthermore, we construct the personalized dynamic vocabulary for each given review by filtering the high-frequency words from the historical text.

The hyper-parameters in our model are tuned from the validation dataset. We set the head number h in multi-head attention to 4. We set encoder layer number N_1 to 6 and decoder layer number N_2 to 6

Table 3: ROUGE performance on the five datasets. The improvements of our proposed method over all baselines are significant with p-value< 0.05.

Dataset	Metric	TextRank	S2S+Attn	PGN	HSSC	memAttr	USN	Dual-view	Transformer	**TRNS**
Toys	Rouge-1	3.98	14.71	15.40	14.77	16.97	15.54	15.80	12.81	**18.87**
	Rouge-2	0.78	2.84	4.21	3.98	4.57	3.10	4.85	2.20	**5.93**
	Rouge-L	3.51	14.35	15.19	14.49	16.68	15.23	15.45	12.68	**18.68**
Sports	Rouge-1	7.63	15.26	16.32	15.44	18.58	14.93	16.63	14.52	**19.88**
	Rouge-2	1.54	4.62	5.36	4.08	**7.29**	5.08	5.12	4.03	6.21
	Rouge-L	6.87	15.13	16.15	15.25	18.39	14.81	16.30	14.39	**19.69**
Electronics	Rouge-1	3.16	14.46	15.60	14.87	18.59	16.94	16.05	15.27	**19.97**
	Rouge-2	0.71	3.24	4.59	4.01	6.30	5.55	5.08	4.28	**7.83**
	Rouge-L	3.15	14.11	15.34	14.35	18.25	16.68	15.94	15.12	**19.63**
Home	Rouge-1	3.79	13.90	15.25	14.34	17.35	13.33	15.43	13.52	**18.21**
	Rouge-2	0.78	4.10	4.47	4.25	6.25	2.91	5.08	2.22	**5.85**
	Rouge-L	3.36	13.74	15.12	14.02	17.16	13.19	15.20	13.25	**18.05**
Movie	Rouge-1	3.16	11.55	12.59	12.32	13.71	13.59	13.06	10.59	**15.49**
	Rouge-2	0.52	2.90	3.82	3.54	4.27	4.11	3.78	2.12	**5.07**
	Rouge-L	2.78	11.29	12.21	12.05	13.31	13.22	12.73	10.27	**15.10**

in our method and the naive transformer. The dimension d_e for the encoder and the decoder in our model and transformer is set to 512 (tuning in [128,256,512,1024]). The dimension d_{ff} for feed-forward network in all encoder layers and decoder layers is set to 2048. The dimension d_w for word embedding and dimension d_c for rating, user and product ID embedding are both set to 300 (tuning in [200, 300, 400]) and we use dropout with probability 0.3 for all datasets (tuning in [0.1, 0.3, 0.5, 0.7]) to avoid overfitting. The RNN in the encoder module is 2-layer GRU. We use the Adam [13] optimizer to train our model. For the parameters in the training, we set the batch size to 48. The student t-test is applying in our experiments to conduct the statistical significance.

5 RESULT AND DISCUSSION

5.1 Performance Evaluation

To evaluate our model, we conduct summary generation for product reviews, the results are list in table 3. Our proposed model TRNS almost achieves the best performance on the ROUGE metric on the five datasets. We have the following observations from the experimental results.

First, our model obtain enormous improvements than extractive method *TextRank* which generates the summary by extracting sentences directly from the given review. Second, our model outperforms the naive *Transformer*, *Seq2Seq+Attn* and *PGN* models with a larger margin. The reason is that our method not only considers the given review but also leverages the helpful personalized attributes along with the given review (e.g., historical summaries) to strengthen the summary generation.

Third, our model achieves better performance than *HSSC* and *Dual-View* even though they also utilize ratings to control the sentiment tendency of the generated summaries. This is because that we further integrate historical summaries of the corresponding user and product to learn more comprehension review representation and design a personalized vocabulary which contains user preferences and product characteristics.

Fourth, our method performs better than *USN* which generates summaries by leveraging the user and product information. It is because that our model model leverages the external historical summaries to learn more comprehension review representation, while *USN* only models the given review text. Especially, *USN* uses the user embedding to select the important information in the encoder, but the user embedding conveys little information and leads to an unstable performance on different datasets. In Table 3, it is obvious that *USN* performs well on some datasets (e.g., "Movie" dataset), while performs poorly on other datasets (e.g., "Home" dataset).

Finally, our model also achieve better or competitive performance than *memAttr* which also incorporates the historical reviews and summaries. The main reason is that our model infers the useful information from the historical summaries by capturing the relevance between the given review and the corresponding historical summaries. Besides, we design a historical reasoning memory and incorporate it into a decoder through the self-attention mechanism to improve the generated summaries quality.

In summary, our model performs better than baselines and the results demonstrate the effectiveness of our model.

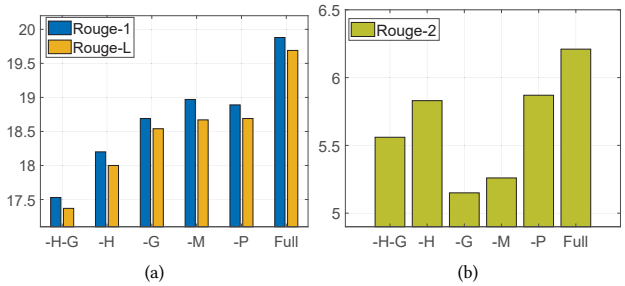

Figure 4: Ablation experiments on the dataset *Sports*. Figure (a) shows the result on ROUGE-1 and Rouge-2 metrics, and Figure (b) shows the result on Rouge-L metric.

5.2 Ablation Study

We conduct ablation experiments on the *Sports* dataset to verify the effect of each important component of our method. We evaluate four main components of our method:

1) "H" denotes the inter-attention between the given review and the corresponding historical summaries;

2) "G" denotes the intra-attention over the given review with personalized attributes as query;

3) "M" denotes historical reasoning memory for input reviews;

4) "P" denotes pointer network that generates words from the memory. The results are reported in the Figure 4.

In the encoder module, "-H" represents that we remove the inter-reasoning attention in the encoder layer. In fact, this model conducts intra-reasoning attention directly over the input review with personalized attributes (i.e., rating, user and product ID) as the query. It is obvious that inferring the useful information from the historical summaries is beneficial to learn more comprehensive review representation and achieve significantly better performance. Then, "-G" represents that we remove the intra-attention in each encoder layer. The results show that the less salient or irrelevant contents lead to the model can not focus on the contents that should be included in the target summary. In addition, "-H-G" denotes that we remove the inter- and intra- reasoning attention simultaneously, i.e., only reserving the review encoding in the encoder. We can observe that this model performs poorly on all ROUGE metrics.

In the decoder module, "-M" denotes that the historical reasoning memory and the memory-decoder attention over it are both removed from the decoder. We can see that the historical memory contributes to better performance. Then, "-P" denotes that we remove the pointer network after the decoder layer stack and directly generate target words from the vocabulary. As shown in Figure 4, we can find that the historical reasoning memory is helpful to the effectiveness of our model.

Table 4: Performance of our model on *Sports* dataset w.r.t a different number of stack layers in encoder and decoder.

Encoder	ROUGE-1	ROUGE-2	ROUGE-L
2-layer	19.39	5.60	19.24
4-layer	19.19	5.95	18.93
6-layer	**19.88**	6.21	**19.69**
8-layer	19.30	**6.77**	19.08
Decoder	ROUGE-1	ROUGE-2	ROUGE-L
2-layer	18.90	5.06	18.59
4-layer	19.36	5.86	19.20
6-layer	**19.88**	6.21	**19.69**
8-layer	19.42	**6.65**	19.00

5.3 Parameter Analysis

Since the transformer is our base model, hence in this section, we conduct experiments to analyze the performance under different stacked layers. In detail, we adjust the layers of the encoder and decoder to 2, 4, 6, and 8 respectively, and the results are listed in Table 4. We fix the decoder (encoder) to 6 layers when adjusting the encoder (decoder) structure. The results demonstrate that employing 2, 4 layers for either encoder or decoder makes the model converges quickly, while performs worse than other parameter settings. When employing 8 layers, the model needs more time to converge and achieves little improvements. In order to balance the convergence time and effectiveness, we employ a 6 layers stack structure for both encoder and decoder on all datasets.

5.4 Discussion

Since the core of our proposed method is to effectively incorporate the history summaries into review summary generation, in this section, we further step into the inside of our model to discuss different strategies using the user/product information. Particularly, we design several model variants and conduct experiments on *Sports* dataset in the following three aspects.

Firstly, we analyze the effect of different ways of integrating the historical summaries in the encoder part. Two model variants of our model are designed:

1) **TRNS-I**: simply integrating historical summaries embeddings $\mathbf{Z}$ as context vector into the intra-reasoning attention module i.e., $\mathbf{q} = MLP([\mathbf{u}; \mathbf{p}; \mathbf{r}; \mathbf{Z}])$ in Equation (10);

2) **TRNS-C**: concatenating historical summaries embeddings and words embeddings to represent input reviews in Section 3.3.1.

Both the two models are without the inter-reasoning attention module. The experimental results are listed in Table 5. It shows that our model with inter-reasoning attention achieves the best performance. TRNS-I only leverages the historical summaries to select the informative words from the input review, and completely ignores the rich information in historical summaries. TRNS-C simply combining the features of historical summaries could not capture the important interaction between the historical information and the input review, which leads to the performance decline. Our model would infer the useful information from historical summaries given the input review through the inter-reasoning attention module.

Table 5: Performance comparison.

	ROUGE-1	ROUGE-2	ROUGE-L
TRNS-I	18.81	5.07	18.64
TRNS-C	19.21	5.61	18.97
TRNS	**19.88**	**6.21**	**19.69**

Secondly, we explore the effectiveness of intra-reasoning attention module in the encoder. It should be noted that intra-reasoning attention in our model aims to select important words at each encoder layer. We design another variant model **TRNS-S** that only conducts important contents selection after the N_1 encoder layers instead of at each encoder layer. The results are listed in Table 6. We can see that the our model TRNS performs better. This is mainly because that (1) intra-reasoning attention at each encoder could fully focus on important words in the review compared with the TRNS-S; (2) the incorporation of the intra-reasoning and inter- reasoning attention could enhance each other through the propagation of information between the encoder layers.

Finally, we conduct experiments to evaluate the effectiveness of the historical reasoning memory in the decoder module, i.e., the specific dynamic vocabulary constructed from the historical

Table 6: Performance comparison.

	ROUGE-1	ROUGE-2	ROUGE-L
TRNS-S	19.16	5.00	18.82
TRNS	**19.88**	**6.21**	**19.69**

summaries. We design two other strategies to generate summaries without the above dynamic vocabulary:

1) **TRNS-NC**: generating summaries only from the common vocabulary.

2) **TRNS-R**: generating summaries from the common vocabulary and copy words from the input review.

The results are listed in Table 7. We can see that TRNS-NC performs worse than other models without the copy mechanism. It is obvious that our model TRNS achieves better performance than TRNS-R. This is mainly because that the specific vocabulary constructed from the historical text contains more out-of-vocabulary words than the input reviews. And the historical reasoning memory is beneficial to generate more personalized target words by reasoning words from the constructed vocabulary.

Table 7: Performance comparison.

	ROUGE-1	ROUGE-2	ROUGE-L
TRNS-NC	18.77	4.83	18.54
TRNS-R	19.20	5.02	19.04
TRNS	**19.88**	**6.21**	**19.69**

5.5 Visualization of Attention

To demonstrate that our model can infer the relevant words from the historical summaries with the input review as the query, we conduct the visualization of attention. We select a short review and the corresponding historical summaries. Figure 5 shows the attention weights of the inter-attention sub-layer in the last encoder layer. We can see that there is significantly different relevance of different words of the historical summaries sequence towards the same word of the input review. For example, "better" of the input review has higher weights with "fun" than "disappointment" of the historical summaries. It is obvious that "disappointment" is irrelevant to the main content of the input review.

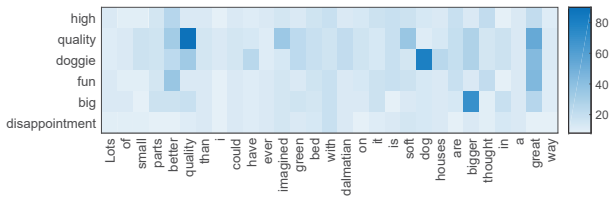

Figure 5: Visualization of attention weights between the historical summaries and the input review.

5.6 Case Study

We further conduct a case study to analyze the quality (e.g., readability) of the generated summaries. The results are listed in Figure 6.

We can conclude that the generated summary is very readable and grammatically correct. In the first case, our method captures the main content "great quality". Compared with the ground-truth summary "cute", the generated summary not only represents the positive sentiment tendency of the user and the important aspect (i.e., "quality") of the product. Thus, the generated summary is more helpful to both users and business. In the second case, we can see that our model could capture more comprehensive content in the input review. That is the quality of the computer is good but the user is not satisfied with the video play. In the last case, our method describes the product as "great for learning letters", which is the significant characteristic of the product (i.e., "puzzle"). Some of the historical summaries mention this characteristic, for example, "great letter learning puzzle" from another user to this product. In conclusion, our reasoning model can better infer the important information by considering the relevance between the input review and the corresponding historical summaries .

Review: Lots of small parts better quality than i could have ever imagined. The green bed with the dalmatian on it is soft. The dog houses are bigger than i thought in a great way.
Transformer: Cute.
TRNS: Great quality.
Ground-truth summary: Amazing.

Review: This is a great computer but I'm taking one star off because video play back is not as good as in cheap windows dell i3 laptop. Dell i3 laptop uses same integrated video card ||…
Transformer: Great quality.
TRNS: Great computer but video is not as good as i expected.
Ground-truth summary: Video could be better.

Review: The colorful chunky letters are very appealing. The fact that they are not level with each other is very freeing to my grandson and he's learned more words with this. This very durable puzzle will last for generations.
Transformer: Great puzzle.
TRNS: Great puzzle for learning letters.
Ground-truth summary: Great abc puzzle.

Figure 6: Several generated summaries and the corresponding review, reference summary.

6 CONCLUSION

In this paper, we propose a transformer-based reasoning network for personalized review summarization, which is able to effectively incorporate the historical summaries into the review summary generation. The core of our method is to fully capture the complicated relevance between the given review and historical summaries via our well-designed reasoning modules both in encoder and decoder. In the encoder, we design an inter- and intra-attention in each encoder layer to select more informative parts of historical summaries in terms of the input review and learn more comprehensive representation. In the decoder, we construct a historical reasoning memory from the historical summaries and generate the out-of-vocabulary target words from it. The experimental results on five commonly used datasets demonstrate that our method achieves better performance than the state-of-the-art methods.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2020YFC0833303), and the National Natural Science Foundation of China (61902278).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

[2] Giuseppe Carenini, Jackie Chi Kit Cheung, and Adam Pauls. 2013. Multi-document summarization of evaluative text. *Computational Intelligence* 29, 4 (2013), 545–576.

[3] Hou Pong Chan, Wang Chen, and Irwin King. 2020. A Unified Dual-View Model for Review Summarization and Sentiment Classification with Inconsistency Loss. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. Association for Computing Machinery, 1191–1200.

[4] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[5] Giuseppe Di Fabbrizio, Amanda Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*. 54–63.

[6] 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: Volume 1, Long Papers*. 623–632.

[7] Xiangyu Duan, Hongfei Yu, Mingming Yin, Min Zhang, Weihua Luo, and Yue Zhang. 2019. Contrastive Attention Mechanism for Abstractive Sentence Summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3035–3044.

[8] Kavita Ganesan, Cheng Xiang Zhai, and Jiawei Han. 2010. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *23rd International Conference on Computational Linguistics, Coling 2010*.

[9] Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-Up Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4098–4109.

[10] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1602–1613.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[12] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 168–177.

[13] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015 : International Conference on Learning Representations*.

[14] Junjie Li, Haoran Li, and Chengqing Zong. 2019. Towards personalized review summarization via user-aware sequence network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6690–6697.

[15] Junjie Li, Xuepeng Wang, Dawei Yin, and Chengqing Zong. 2019. Attribute-aware Sequence Network for Review Summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2991–3001.

[16] Piji Li, Lidong Bing, Zhongyu Wei, and Wai Lam. 2020. Salience Estimation with Multi-Attention Learning for Abstractive Text Summarization. *arXiv preprint arXiv:2004.03589* (2020).

[17] Pan Li and Alexander Tuzhilin. 2019. Towards Controllable and Personalized Review Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3228–3236.

[18] Piji Li, Zihao Wang, Lidong Bing, and Wai Lam. 2019. Persona-Aware Tips Generation. In *The World Wide Web Conference*. ACM, 1006–1016.

[19] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *SIGIR*. ACM, 345–354.

[20] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[21] Junyang Lin, Xu Sun, Shuming Ma, and Qi Su. 2018. Global Encoding for Abstractive Summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 163–169.

[22] Hui Liu and Xiaojun Wan. 2019. Neural Review Summarization Leveraging User and Product Information. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2389–2392.

[23] Hongtao Liu, Fangzhao Wu, Wenjun Wang, Xianchen Wang, Pengfei Jiao, Chuhan Wu, and Xing Xie. 2019. NRPA: neural recommendation with personalized attention. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1233–1236.

[24] Yang Liu and Mirella Lapata. 2019. Hierarchical Transformers for Multi-Document Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5070–5081.

[25] Shuming Ma, Xu Sun, Junyang Lin, and Xuancheng Ren. 2018. A hierarchical end-to-end model for jointly improving text summarization and sentiment classification. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 4251–4257.

[26] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

[27] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1073–1083.

[28] Yufei Tian, Jianfei Yu, and Jing Jiang. 2019. Aspect and opinion aware abstractive review summarization with reinforced hard typed decoder. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2061–2064.

[29] 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 *Advances in neural information processing systems*. 5998–6008.

[30] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems* 28 (2015), 2692–2700.

[31] Lu Wang and Wang Ling. 2016. Neural Network-Based Abstract Generation for Opinions and Arguments. In *Proceedings of NAACL-HLT*. 47–57.

[32] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[33] Xianchen Wang, Hongtao Liu, Peiyi Wang, Fangzhao Wu, Hongyan Xu, Wenjun Wang, and Xing Xie. 2019. Neural review rating prediction with hierarchical attentions and latent factors. In *International Conference on Database Systems for Advanced Applications*. Springer, 363–367.

[34] Wenting Xiong and Diane Litman. 2014. Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers*. 1985–1995.

[35] Min Yang, Qiang Qu, Ying Shen, Qiao Liu, Wei Zhao, and Jia Zhu. 2018. Aspect and sentiment aware abstractive review summarization. In *Proceedings of the 27th international conference on computational linguistics*. 1110–1120.

[36] Yongjian You, Weijia Jia, Tianyi Liu, and Wenmian Yang. 2019. Improving abstractive document summarization with salient information modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2132–2141.

[37] Wei Zhang, Yue Ying, Pan Lu, and Hongyuan Zha. 2020. Learning Long-and Short-Term User Literal-Preference with Multimodal Hierarchical Transformer Network for Personalized Image Caption.. In *AAAI*. 9571–9578.

[38] Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5059–5069.