Knowledge-Based Systems 218 (2021) 106858

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Rating-boosted abstractive review summarization with neural personalized generation

Hongyan Xu [a], Hongtao Liu [a], Wang Zhang [a], Pengfei Jiao [d], Wenjun Wang [a,b,c,*]

[a] College of Intelligence and Computing, Tianjin University, Tianjin, China
[b] State Key Laboratory of Communication Content Cognition, Beijing, China
[c] College of Information Science and Technology, ShiHezi University, Xinjiang, China
[d] Center for Biosafety Research and Strategy, Law School, Tianjin University, Tianjin, China

ARTICLE INFO

Article history:
Received 26 September 2020
Received in revised form 31 December 2020
Accepted 9 February 2021
Available online 12 February 2021

Keywords:
Summary generation
Rating prediction
Recommender system

ABSTRACT

In this paper, we study abstractive summarization for product reviews in the recommender systems, which aims to generate condensed text for online reviews. The summary generation is not only relevant with the content of the review itself but should be fully aware of the intrinsic features of the corresponding user and product, i.e., personalization, which are helpful to identify the saliency information in the reviews. Therefore, we propose a Rating-boosted Abstractive Review Summarization with personalized generation (RARS). In our approach, we first propose a neural review-level attention model to effectively learn user preference embedding and product characteristic embedding from their history reviews. Then, we design a personalized decoder to generate the personalized summary, which utilizes the representations of the user and the product to calculate saliency scores for words in the input review to guide the summary generation process. In addition, the rating information can explicitly indicate the sentiment opinion, hence we jointly optimize the summary generation and rating prediction through a multi-task framework, where the two tasks inherently share user preference embedding and product characteristics embedding. Extensive experiments on four datasets show that our model can effectively improve the performance of both review summarization and rating prediction.

1. Introduction

Abstractive review summarization aims to condense the product review of recommendation system to a shorter version, which contains the main information of the original review. With the rapid development of e-commerce, review summarization has attracted more and more attention for the ability to save time for users to browse the reviews and help users make purchase decisions quickly [1].

In recent years, text summarization has been widely studied in the natural language processing field. For example, See et al. [2] propose a pointer generator network (PGN) which allows both copying words from the input text and generating words from the fixed vocabulary. However, compared with the conventional text summarization, product reviews in recommender systems contain rich extra information, which can be utilized to improve the quality of the generated summaries, such as history reviews of users and products, rating information, etc. Many works have been proposed in this field. Ma et al. [3] propose a model that jointly improves text summarization and sentiment classification by treating the sentiment label as the further "summarization" of the text summarization output. Li et al. [1] propose a user-aware model that incorporates the user characteristics into summary generation because different users care about different aspects towards the same review.

Despite their great performance in review summarization, there leaves a lot to be desired. Firstly, user preferences and product characteristics are beneficial to generate personalized summaries, since different users and products may lead to different summaries even if the reviews are similar. Secondly, the naive concatenation of the history reviews into a single document is suboptimal for user/product representation learning in most existing methods. In fact, different history reviews are differently important in learning user preference embedding and product characteristic embedding in terms of abstractive review summarization. In addition, when users write reviews to the target

The code (and data) in this article has been certified as Reproducible by Code Ocean:https://help.codeocean.com/en/articles/1120151-code-ocean-s-verification-process-for-computational-reproducibility. More information on the Reproducibility Badge Initiative is available at www.elsevier.com/locate/knosys.

* Correspondence to: Peiyang Park Campus, No. 135 Yaguan Road, Haihe Education Part, Tianjin, China.
E-mail addresses: hongyanxu@tju.edu.cn (H. Xu), htliu@tju.edu.cn (H. Liu), wangzhang@tju.edu.cn (W. Zhang), pjiao@tju.edu.cn (P. Jiao), wjwang@tju.edu.cn (W. Wang).

https://doi.org/10.1016/j.knosys.2021.106858
0950-7051/

product, rating information is simultaneously given and explicitly reflects the sentiment opinion of the users. Therefore, it is intuitive that the rating information along with the input review is highly relevant with summary generation since the generated summary should have the same sentiment tendency with the input review.

Based on the above observation, we propose a Rating-boosted Abstractive Review Summarization with personalized generation (RARS). Our method mainly contains three modules: user/product encoder, summary decoder with personalized information, and a jointly training framework. In the user/product encoder, reviews are firstly fed into a bidirectional GRU [4] to get their semantic representations, following a word embedding layer. In contrast to the attention of previous works which use general query vectors for all reviews, we design a review-level attention module to capture the different contributions of history reviews for the user preference and product characteristic modeling. In detail, we leverage the current product ID embedding to select the informative reviews from the user history reviews to learn the user preference, and learn the product characteristic embedding with the user ID embedding as the query. Then, the user and product embeddings learned in the encoder are fed into the summary decoder to enhance the summary generation. In addition, we employ a multilayer perceptron over the learned user and product embeddings to get the personalized feature representation, which will be used to conduct rating prediction.

In the decoder module, we first calculate the saliency score for each word in the input review by utilizing the user preference and the product characteristic embeddings as the query vectors. Then, we use the saliency score to re-weight the attention between the input review and the decoder, allowing our model can focus more on words that users prefer or reflect the product features. Finally, we define a joint objective function to optimize the summary generation task and rating prediction task, and these two tasks inherently share the user preference embedding and the product characteristic embedding. It allows that the summary generation is guided by the rating information and the rating prediction is improved by identifying the important content in the reviews.

Overall, our contributions are as follows: (1) we propose a unified and effective abstractive review summarization algorithm which learns the user preference embedding and product characteristic embedding with considering the different usefulness of the history reviews of users and products. (2) we design a personalized decoder which calculates the saliency score based on user preferences and product characteristics, allowing the model to identify the key words in the input review. (3) experimental results show that our model could achieve better performance than baselines both in summary generation and rating prediction.

2. Related works

Summarization for product reviews. Text summarization is a significant branch in natural language processing tasks, which aims to generate short, concise text for the given text and has made considerable progress recently [5–8]. There are two broad approaches for summarization: extractive methods that select the salient component from the input text as summaries [7–9], and abstractive methods that generate words one by one like human writing summaries [2,6]. Especially, See et al. [2] propose the copy mechanism, which uses a probability to decide to generate words from the vocabulary or copy words from the source text.

In the field of recommender system, review text usually has personalized information (e.g., user ID, product ID, and ratings) which plays a crucial role in the review summarization task. Early approaches for review summarization are extractive [10–12], which extract words from reviews as summaries. Xiong et al. [12] propose an unsupervised extractive method for online reviews summarization by using the helpfulness ratings for review-level filtering and as the supervision of the topic model for sentence-level content scoring. Previous works have shown that abstractive methods perform better than extractive methods on review summarization task [13,14]. Thus, we mainly focus on the abstractive summarization methods in this paper. In the area of recommender system , LSTM [15] or GRU [4] based encoder–decoder frameworks have been widely used in review generation [16,17] and summary generation [12,18–24].

Recently, some approaches propose to generate summaries for reviews with considering user preferences and product characteristics. Li et al. [22] propose to generate summaries for reviews by considering the user attributes (e.g., gender, age, occupation) and writing styles. Liu et al. [23] select some similar history reviews as the memory for the user and product of the input review and aggregate this information as a context vector into the decoder to generate the personalized summary. In addition, there are methods that utilize the multi-task learning framework to jointly integrate other tasks with review summarization. Ma et al. [3] jointly improve the sentiment classification and product review summarization tasks by treating the sentiment label as the further "summarization" of the review summarization output. Hou et al. [25] also propose a dual-view model to jointly improve the sentiment classification and summarization tasks, and introduce an inconsistency loss to penalize the disagreement between the source-view and summary-view sentiment classifiers. The most similar works with our method are [21,24], which conduct user and product latent factor modeling, rating prediction, and generate summary based on the learned user and product latent factors. However, we further consider the importance of different history reviews to effectively learn the user preference embedding and the product characteristic embedding.

Rating prediction. For rating prediction task, traditional methods based on Matrix Factorization (MF) [26] have been studied for a long time and there have been various methods, such as PMF [27], NMF [28], SVD++ [29]. However, these methods perform poorly when the rating is very sparse. So, some other approaches [30,31] propose to apply the topic model on review text to predict the product ratings more accurately. Recently, neural network based approaches outperform the traditional baselines by utilizing the reviews [21,32–34], the information of users and products [35–39] to improve the recommender system performance. Chen et al. [32] propose to conduct rating regression with considering the usefulness of the reviews by introducing an attention mechanism. Different from them, we propose a multi-task framework, where the summary generation task and rating prediction task inherently share the user preference embedding and the product characteristic embedding and optimize the two tasks simultaneously.

3. Proposed method

In this section, we introduce the proposed method. The overview of our approach is shown in Fig. 1, where the left part is user preferences and product characteristics modeling with review-level attention, the right part is summary generation with personalized saliency score calculation. In addition, a multi-task learning framework is designed to jointly optimize both review summarization and rating prediction.

3.1. Problem formulation

We formally define the problem of review summarization and rating prediction as follows. Given the input review $\mathbf{X} = [w_1, w_2, \ldots, w_m]$, where m is the length of the review, and the

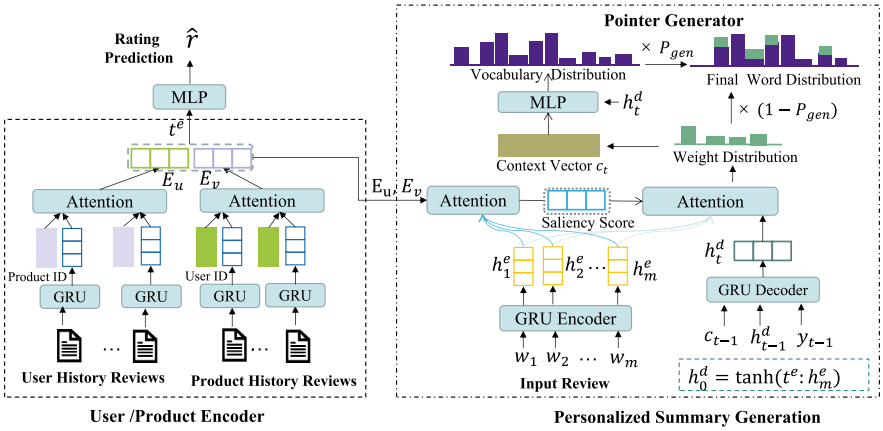

Fig. 1. The framework of our *RARS* approach.

corresponding personalized information, (i.e., the user id u and product id v), the review summarization task aims to generate a summary for the given review. The generated summary is denoted as $\hat{\mathbf{Y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_l\}$ and the reference summary is denoted as $\mathbf{Y} = \{y_1, y_2, \ldots, y_l\}$, where l is the length of the summary. The rating prediction task aims to predict the rating score that user u would give to product u.

The history reviews could help learn more comprehensive representations for user preferences and product features, which are shared between review summarization and rating prediction. Thus, we construct a history review set for the user u by randomly selecting K history reviews of the user $S_u = [(v_1, X_1), \ldots, (v_K, X_K)]$, where X_i represents the history review and v_i represents the corresponding product ID. In the same way, we construct history review set for the product v, which is denoted as $S_v = [(u_1, X_1), \ldots, (u_K, X_K)]$, where X_i represents the history review and u_i represents the corresponding user ID. We discuss the selection of K in Section 5.

3.2. User preference and product characteristic modeling

In this section, our method designs a user/product encoder to learn the representations for users and products from the history reviews. In detail, we first use a GRU-based review encoder to learn semantic representation for history reviews of users/products and then adopt review-level attention to select the more informative reviews. Moreover, the predicted rating score is calculated by applying non-linear transformation on the concatenation of the learned user and product representations.

3.2.1. GRU-based review encoder

Given a review $X = [w_1, w_2, \ldots, w_m]$, the encoder produces a sequence of hidden states $[\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_m}]$ and m is the length of the review. In this paper, we employ the bi-directional GRU to get the hidden states: $\mathbf{h_t} = [\overrightarrow{h_t} : \overleftarrow{h_t}]$, where $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ represent the hidden states in forward and backward GRU respectively, and [:] is concatenation operation. Then, the representation of the review is denoted as $\mathbf{h} = [\overrightarrow{h}_{\mathbf{m}} : \overleftarrow{h}_{\mathbf{m}}]$.

3.2.2. User/product encoder with review attention

Since the importance of different reviews are different for the user preference and the product characteristic modeling, we adopt a review-level attention network to select more informative reviews from the history reviews.

We first define the ID embeddings of users and products as $\mathbf{U} \in \mathbb{R}^{d \times |\mathcal{U}|}$ and $\mathbf{P} \in \mathbb{R}^{d \times |\mathcal{V}|}$ respectively, where $|\mathcal{U}|$ is the number of users, $|\mathcal{V}|$ is the number of products and d is the embedding dimension. ID embeddings are widely used in recommender systems and can be viewed as the latent features of users and products to indicate their inherent properties [32]. Given the history review set of the user $S_u = [(v_1, X_1), \ldots, (v_K, X_K)]$, this module learns the user preference representation by aggregating the history reviews via an attention network. We calculate the attention scores a_i for the ith review in S_u as following:

$$a_i = f(\mathbf{W}_u^T tanh(\mathbf{W}_{av}\mathbf{v}_i + \mathbf{W}_{ah}\mathbf{h}_i)) , \tag{1}$$

where f is the softmax function, $\mathbf{W}_{av}$, $\mathbf{W}_{ah}$ and $\mathbf{W}_u$ are the attention parameters. $\mathbf{v}_i \in \mathbf{P}$ is the ID embedding of the history product v_i, and $\mathbf{h}_i$ is the feature vector of history review x_i learned by the review encoder in Section 3.2.1. Then the final user representation $\mathbf{E}_u$ is denoted as the weighed summation of all the history reviews in S_u:

$$\mathbf{E}_u = \sum_{i=0}^{K} a_i \mathbf{h}_i . \tag{2}$$

Likewise, we get the characteristic embedding $\mathbf{E}_v$ for product v. Subsequently, the interaction feature $\mathbf{t}^e$ between user u and product v can be derived from the combination of $\mathbf{E}_u$ and $\mathbf{E}_v$, denoted as:

$$\mathbf{t}^e = f([\mathbf{E}_u : \mathbf{E}_v]) , \tag{3}$$

where f is a linear projection. Afterward, the representation $\mathbf{t}^e$ for the user and product will be applied for the following personalized summary generation and rating prediction under a multi-task learning framework.

3.2.3. Rating prediction

Ratings are integers given by users along with reviews and are quite related to the summary generation. In this part, we conduct rating prediction as the auxiliary task via using the personalized feature vector $\mathbf{t^e}$ which is learned from the user and product information. The predicted real-valued rating $\hat{r}$ is calculated based on $\mathbf{t}^e$ using a multi-perceptron layer:

$$\hat{r} = \mathbf{W}_{re} ReLU(\mathbf{t}^e) + b_r , \tag{4}$$

where $ReLU(x) = max(0, x)$ is a non-linear function. $\mathbf{W}_{re}$ and b_r are model parameters, and would be randomly initialized and are updated during the model training phase.

3.3. Personalized summary generation

After obtaining the representations of the user and product, this section aims to generate personalized summary for the input review. We adapt the traditional Pointer-Generator Neural

Network (PGN) [2], which not only generates words from the vocabulary but also copy words from the input text. In particular, we utilize the personalized representations from Section 3.2.2 to conduct the salient content selection in the decoder.

Given the current review, we first use the GRU-based review encoder in Section 3.2.1 to obtain the hidden state sequence for the input review, denoted as $[\mathbf{h}_1^e, \mathbf{h}_2^e, \ldots, \mathbf{h}_m^e]$. Then, our personalized decoder is initialized by the concatenation of the last hidden vector $\mathbf{h}_m^e$ and the personalized interaction feature $\mathbf{t}^e$ for user u and product v:

$$\mathbf{h}_0^d = tanh([\mathbf{h}_m^e : \mathbf{t}^e]) . \tag{5}$$

In this way, the characteristics of users and products can be integrated into the summary generation.

Subsequently, the decoder generates a summary for the input review with one word a step. Based on the embedding of the output word at the previous step $\mathbf{y}_{t-1}$, decoder hidden state $\mathbf{h}_{t-1}^d$ and the context vector $\mathbf{c}_{t-1}$, the hidden state $\mathbf{h}_t^d$ can be calculated as follows:

$$\mathbf{h}_t^d = GRU(\mathbf{y}_{t-1}, \mathbf{h}_{t-1}^d, \mathbf{c}_{t-1}) . \tag{6}$$

At time step t, the alignment between the encoder and decoder is highly relevant to the personalized information. Therefore, we design a personalized attention mechanism which is able to focus more on important words that reflect the preference of user u and the characteristic of product v. First, the saliency score is calculated based on the relationship among user preference embedding $\mathbf{E}_u$, product characteristic embedding $\mathbf{E}_v$ and all the encoder hidden states $\mathbf{h}_i^e$.

$$\beta_i = \sigma(\mathbf{E}_u \cdot \mathbf{h}_i^e + \mathbf{E}_v \cdot \mathbf{h}_i^e) \tag{7}$$

where σ is the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$, and $\beta_i \in (0, 1)$ is the personalized score for word i in the input review.

Furthermore, we incorporate the calculated saliency score into the attention network. Hence, the attention distribution over the input review words $\alpha_{t,i}$ can be calculated as follows:

$$\alpha_{t,i} = Softmax(\beta_i \mathbf{Z}_\alpha^T tanh(\mathbf{W}_\alpha[\mathbf{h}_t^d : \mathbf{h}_i^e])) , \tag{8}$$

where, $\mathbf{W}_\alpha$, $\mathbf{Z}_\alpha$ are learnable parameters. The context vector $\mathbf{c}_t$ is a weighted sum of the input review hidden states $\mathbf{h}_i^e$:

$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \mathbf{h}_i^e . \tag{9}$$

Then, we could get the final vocabulary distribution P_{vocab} by applying a softmax function over the concatenation of decoder state $\mathbf{h}_t^d$ and the context vector $\mathbf{c}_t$:

$$P_{vocab} = Softmax(\mathbf{W}_b tanh(\mathbf{W}_c[\mathbf{c}_t : \mathbf{h}_t^d])) , \tag{10}$$

where $\mathbf{W}_b$, $\mathbf{W}_c$ are model parameters. Following PGN [2], we also use a probability $p_{gen} \in [0, 1]$ to control to copy words from the input review or generate words from the vocabulary with probability distribution P_{vocab}.

$$p_{gen} = \sigma(\mathbf{W}_g[\mathbf{h}_t^d : \mathbf{c}_t : \mathbf{y}_{t-1}] + b_{gen}) , \tag{11}$$

where, σ is the sigmoid function, $\mathbf{W}_g$ and b_{gen} are model parameters.

Finally, the probability distribution $P(\hat{y}_t)$ for the output word at step t is calculated as the weighted sum of the vocabulary distribution P_{vocab} and copy distribution over the input review:

$$P(\hat{y}_t) = p_{gen} P_{vocab} + (1 - p_{gen}) \sum_{i:x_i = y_t} \alpha_{t,i} , \tag{12}$$

where $P(\hat{y}_t)$ is the probability distribution over the extended vocabulary which contains the original vocabulary and all words in the input review.

3.4. Multi-task learning

Summary generation task and rating prediction task share the user preference and product characteristic information, which is beneficial to strengthen the personality of the generated summaries and learn comprehensive user and product representation for better rating prediction. Therefore, we design a multi-task learning framework to optimize both tasks.

For summary generation task, we use the negative log-likelihood as the loss function during training:

$$\mathcal{L}_s = \sum_{t=0}^{l} -logP(\hat{y}_t), \tag{13}$$

where l is the length of the generated summary. For rating prediction task, we formulate the optimization of parameters as a regression problem and the loss function is:

$$\mathcal{L}_r = \frac{1}{|\mathcal{X}|} \sum_{u \in \mathcal{U}, v \in \mathcal{V}} (\hat{r}_{u,v} - r_{u,v})^2 \tag{14}$$

where, $|\mathcal{X}|$ is the size of training data, $\hat{r}_{u,v}$ is the ground truth rating given by user u to product v and $r_{u,v}$ is the predicted rating. Finally, we jointly minimize the losses:

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_r \tag{15}$$

where λ is a hyper-parameter to balance review summarization task and rating prediction task.

The following Algorithm 1 presents the training algorithm for our model. The model parameters, denoted as Θ, are updated by calculating the gradient for them during training procedure. During test, based on the user and product representations in encoder module, our model generates summary for the input review and predicts the rating that the current user might give to the target product simultaneously.

Algorithm 1 The training algorithm for our proposed model.

Input: review dataset $\mathcal{D}$ (which is consist of review text $\mathbf{X}$, user id u, product id v, rating r, and summary text $\mathbf{Y}$)
Output: model parameters Θ

```
Randomly Initialize model parameters Θ
for epoch=0 to N_e do
  for batch=0 to |D|/batchsize do
    Construct history review set for user S_u and product S_v respectively.
    Calculate the user and product representations based on Equation (1-3)
    Calculate the predicted rating r̂ based on Equation (4)
    for t=0 to l do
      Generate word Ŷ_t based on Equation (5-12)
    end for
    Calculate multitask learning loss based on Equation (13–15)
    Update model parameters Θ
  end for
end for
return Θ
```

4. Experimental setup

4.1. Datasets and experimental settings

4.1.1. Datasets

To evaluate our model, we conduct experiments on four datasets from different domains in Amazon[1]: **Electronics**, **Home**

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

Table 1
Overview of the datasets.

	Electronics	Home & Kitchen	Toys & Games	Movie & TV
users	191,522	66,212	19,412	123,960
items	62,333	27,991	11,924	50,052
reviews	1,684,779	550,461	167,504	1,697,471

and Kitchen, **Toys and Games** and **Movies ans TV**. Since the real-world datasets are usually sparse [23], we only reserve the reviews between active users and popular products. Hence, we reserve users and products that have at less K history reviews. In our dataset, each sample contains user ID, product ID, rating, review text, and summary text. The ratings of all datasets are integers and are in the range of [1,5]. The statistics of datasets are shown in Table 1. Following previous work [3], we randomly select 1000 samples from each dataset as valid and test dataset respectively, and the rest samples are used to train the model.

4.1.2. Baselines

We compare our model with several competitive methods which can be divided into two types: text summarization methods and rating prediction algorithms. We use the following approaches as our review summarization baselines.

- LexRank [40]: It is a classical text summarization method, which extracts a sentence as the final summary based on the PageRank algorithm.
- Seq2seq+Attn [41]: It is a sequence to sequence method with an attention mechanism.
- PGN [2]: It is a classical abstractive summarization approach which leverage copy mechanism to alleviate the out-of-vocabulary problem in summary generation based on the pointer network.
- HSSC [3]: It proposes a multi-task framework to jointly optimize the summarization and sentiment classification for product reviews.
- USN [1]: It proposes a personalized review summarization model, which considers the user's writing style and preference on different aspects of the product.
- Dual-View [25]: It is a very-recent review summarization model which conducts the sentiment classification to control the sentiment of the generated summary and introduces an inconsistency loss in training to make the generated summary be consistent with the input review with aspect to the sentiment tendency.

The following methods are employed as our rating prediction baselines.

- PMF [27]: Probabilistic Matrix Factorization introduces the Gaussian distribution to learn the latent factors for users and products.
- NMF [28]: It uses the rating matrix as input to conduct recommendations through the non-negative matrix factorization.
- SVD++ [29]: It extends Singular Value Decomposition with neighborhood models and exploits both explicit and implicit feedback by the users.
- HFT [30]: Hidden Factor as Topics utilizes reviews to improve the performance of the recommender system by incorporating Latent Dirichlet Allocation to discover the hidden topics of the reviews.
- NARRE [32]: It predicts precise ratings by considering the usefulness of the reviews.

For baselines, we use their released source codes or our re-implemented code, and keep the hyper-parameters as denoted in their papers for fair comparison.

Table 2
Summary generation performance comparison on *Home* and *Electronics* datasets.

Dataset	Method	ROUGE-1	ROUGE-2	ROUGE-L
Electronics	LexRank	7.46	1.55	6.63
	Seq2seq+Attn	14.46	3.24	14.11
	PGN	15.60	4.59	15.34
	HSSC	14.87	4.01	14.35
	USN	16.94	5.55	16.68
	Dual-View	16.05	5.08	15.94
	RARS	**18.98**	**5.83**	**18.65**
Home	LexRank	5.71	1.53	4.97
	Seq2seq+Attn	13.90	4.10	13.74
	PGN	15.25	4.47	15.12
	HSSC	14.34	4.25	14.02
	USN	13.33	2.91	13.19
	Dual-View	15.43	**5.08**	15.20
	RARS	**16.67**	4.67	**16.49**

4.1.3. Experimental settings

We use 300-dimension GloVe word embedding [42] pretrained on Amazon reviews to initialize the word embedding in experiments. We randomly initialize other model parameters Θ and adopt Adam Optimizer [43] with a learning rate 0.0001 to optimize the parameters of our model. We use the valid dataset to tune hyper-parameters in our model. We set the hidden sate size of GRU 512 (tuning in [128, 256, 512, 1024]), dropout probability 0.1(tuning in range (0, 0.7] and parameter λ 0.3(tuning in range [0.1, 1]). In addition, we set the epoch number N_e to 10, and the batch size is 16. Our implementation uses the Pytorch framework. We repeat each experiment five times and report the average results in the following parts.

4.1.4. Evaluation metric

For the review summarization task, we use ROUGE [44] as the metric to evaluate the quality of the generated summaries, which is widely used in the fields of text summarization task and text generation task. ROUGE counts the number of overlapping units(i.e., n-gram) between the generated summaries and the reference summaries. The higher ROUGE score indicates better performance. Following previous works [2,3], we report the F_1 scores for ROUGE-1, ROUGE-2 and ROUGE-L.

$$ROUGE_N = \frac{\sum_{S \in References} \sum_{grams \in S} Count_{match}(grams)}{\sum_{S \in References} \sum_{grams \in S} Count(grams)} \tag{16}$$

For the rating prediction task, we evaluate our model with Root Mean Square Error (RMSE) as metric, which is widely used in recommender system. The lower RMSE score indicates that the model performs better and the predicted rating is closer to the ground truth rating. Given the predicted rating $\hat{r}_{u,v}$ and the ground truth rating $r_{u,v}$ between user u and product v, RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{u,v} (r_{u,v} - \hat{r}_{u,v})^2} \tag{17}$$

where N is the number of rating between users and products.

5. Experiment results

In this section, we conduct experiments to evaluate the performance of our proposed method on both review summarization task and rating prediction task.

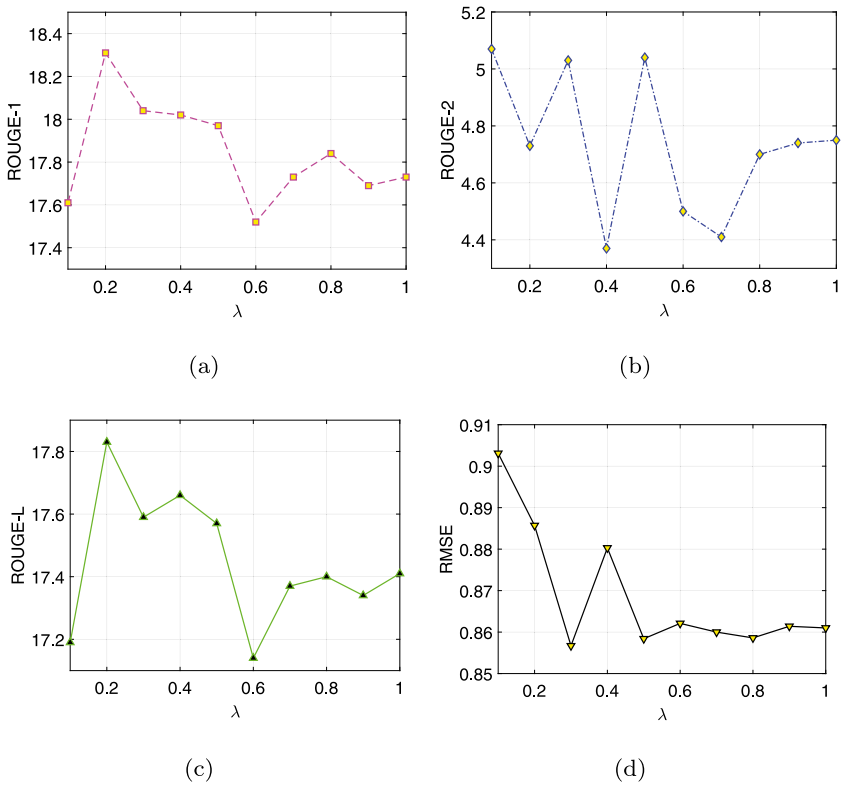

Fig. 2. Performance of our model on *Toys* dataset w.r.t different hyper parameter λ ranging in [0,1]. Figure (a), (b), (c) and (d) correspond to ROUGE-1, ROUGE-2, ROUGE-L and RMSE metrics respectively.

Table 3
Summary generation performance comparison on *Toys* and *Movie* datasets.

Dataset	Method	ROUGE-1	ROUGE-2	ROUGE-L
Toys	LexRank	7.63	1.54	6.87
	Seq2seq+Attn	14.71	2.84	14.35
	PGN	16.08	4.08	15.69
	HSSC	14.77	3.98	14.49
	USN	15.54	3.10	15.23
	Dual-View	15.80	4.85	15.45
	RARS	**18.04**	**5.03**	**17.59**
Movie	LexRank	4.27	1.11	3.72
	Seq2seq+Attn	11.55	2.90	11.29
	PGN	12.59	3.82	12.21
	HSSC	12.32	3.54	12.05
	USN	13.59	4.11	13.22
	Dual-View	13.06	3.78	12.73
	RARS	**15.04**	**5.35**	**14.60**

Table 4
RMSE values for rating prediction.

Method	Electronics	Home & Kitchen	Toys & Games	Movies
PMF	1.553	1.780	1.308	1.307
NMF	1.266	1.220	1.040	1.155
SVD++	1.226	1.164	0.886	1.122
HFT	1.117	1.058	0.893	1.041
NARRE	1.105	1.032	0.881	1.003
RARS	**0.965**	**1.023**	**0.8567**	0.994

5.1. Performance in abstractive review summarization

We first investigate the performance of our model on review summarization task and the results are listed in Tables 2 and 3. We can have the following observations. (1) Compared with the extractive text summarization (i.e., LexRank), our abstractive summarization approach performs much better. The reason may be that summaries for product reviews usually contain words not appearing in the reviews; (2) Our method RARS outperforms the Seq2seq+Attn and PGN which only utilize the input review as input feature. This is because our method explores the rating information to enhance the summary generation, which could explicitly indicates the sentiment of users towards products. (3) Our model achieves better performance than HSSC and Dual-View, which jointly improves summary generation and sentiment classification by leveraging ratings as sentiment label. Because our model learns user preferences and product characteristics from the history review text and integrates this information into the summary generation process, which could be helpful to generate more personalized sentences. (4) It is obvious that our method performs better than USN which also leverages user information to enhance the summary generation. The results show that USN performs well in some datasets, while sometimes performs poorly in other datasets, such as *Home* dataset. Although USN introduces a gate mechanism to select the important words from the source review, it only utilizes the user embedding as the query to calculate the gate weight which does not contain the semantic information. Different from USN, our method calculates the saliency score for each word in the source review by considering the history reviews and the different contributions of them.

In all, our proposed method achieves better performance in the review summary generation than baselines. This validates the effectiveness of our approach.

5.2. Performance in rating prediction

The results are listed in Table 4. First, we can find that our method performs better than traditional methods (e.g., PMF, NMF,

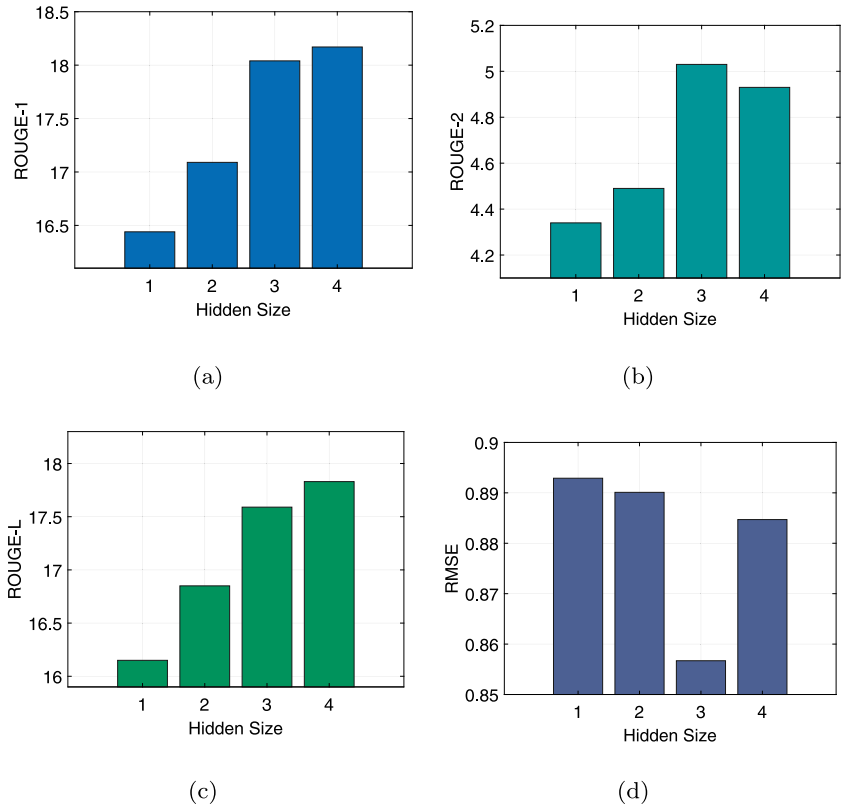

Fig. 3. Performance of our model on *Toys* dataset w.r.t different hidden state size of GRU ranging in [1,5]. Figure (a), (b), (c) and (d) correspond to ROUGE-1, ROUGE-2, ROUGE-L and RMSE metrics respectively.

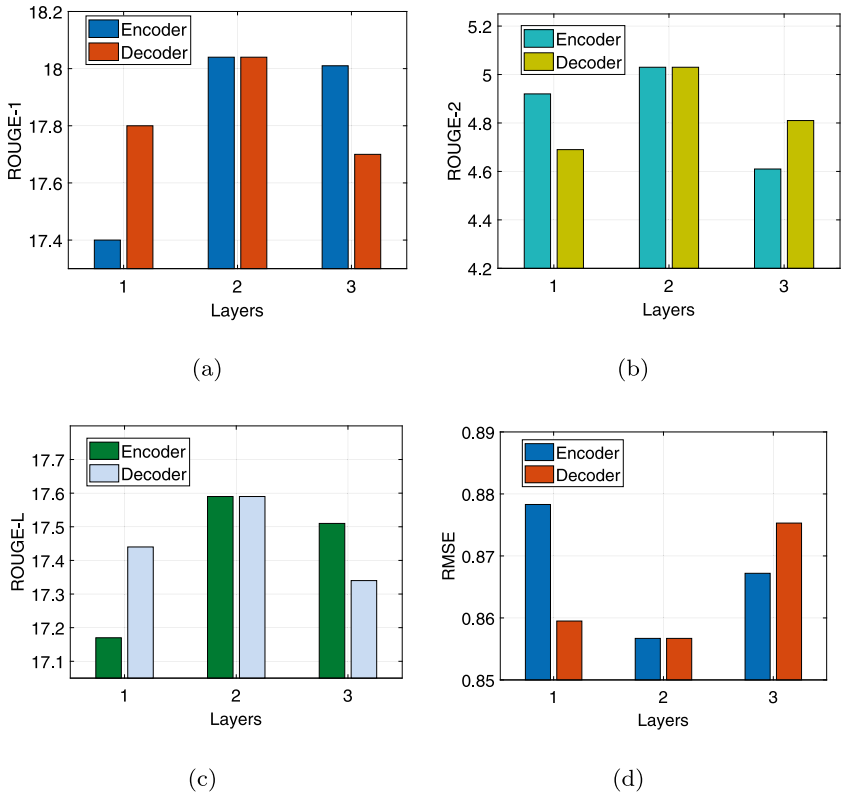

Fig. 4. Performance of our model on *Toys* dataset w.r.t different number of GRU layer, ranging in [1,3], in encoder and decoder. We fix decoder (encoder) to 2 layers when adjust the encoder (decoder) structure. Figure (a), (b), (c) and (d) correspond to ROUGE-1, ROUGE-2, ROUGE-L and RMSE metrics respectively.

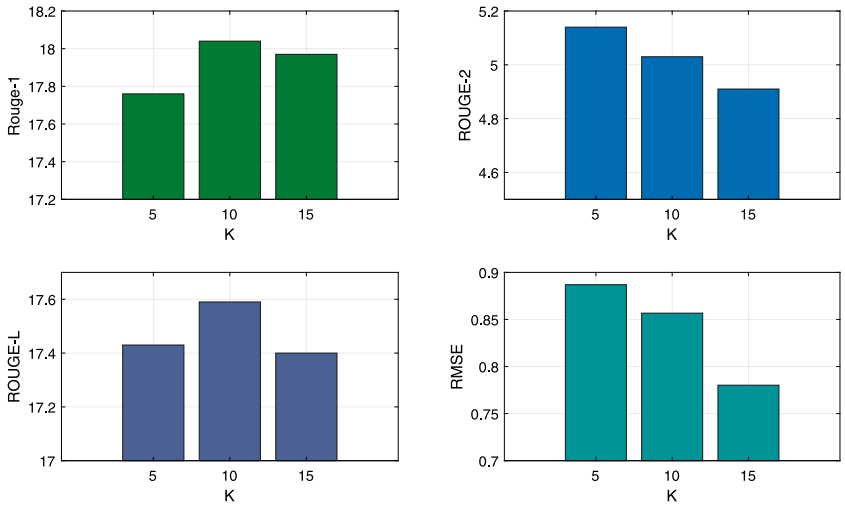

Fig. 5. Performance of our model on *Toys* dataset w.r.t different history review number K. The four sub-figures correspond to ROUGE-1, ROUGE-2, ROUGE-L and RMSE metrics respectively.

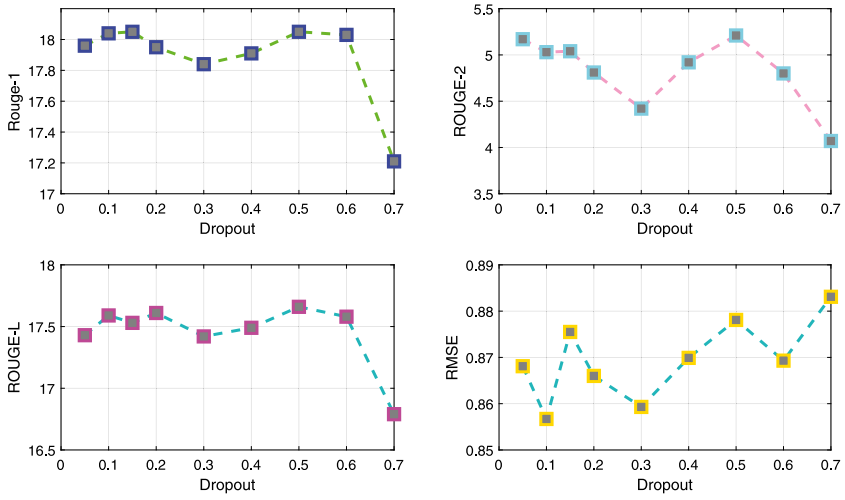

Fig. 6. Performance of our model on *Toys* dataset w.r.t different dropout. The four sub-figures correspond to ROUGE-1, ROUGE-2, ROUGE-L and RMSE metrics respectively.

and SVD++), which only use the rating information. This is because that reviews contain rich information for user preferences and product features. Second, it is obvious that our model outperforms HFT, which incorporates the topic model to discover the hidden topics of the reviews. It is because our method utilizes neural networks to learn the representation for reviews which can get better performance than topic model LDA [45]. Third, our model achieves better performance than NARRE which predicts ratings by considering the usefulness of history reviews. This is mainly because our method develops a multi-task learning framework, in which summary generation task and rating prediction task could enhance each other effectively by sharing the representations of users and products.

5.3. Parameter analysis

Parameters settings in our model affect review summary generation and rating prediction tasks. To decide the default settings, we vary the important parameters to observe how the evaluated metrics changes.

We first investigate the effects of parameter λ towards summary generation and rating prediction tasks. The results are list in Fig. 2. We can see that our model performs best on the summary generation task (e.g., on Rouge-1 and Rouge-L metrics) when λ is 0.2, while it performs poorly on the rating prediction task. However, our model performs best on the rating prediction task (i.e., RMSE metric are lower than others) when λ is 0.3 and it performs well on summary generation. According to this result, we set $\lambda = 0.3$ in all experiments.

Then, we study the effect of the hidden state size of GRU and the number of the GRU layers in the encoder and decoder. The results are list in Fig. 3 and Fig. 4 respectively. Fig. 3 shows that higher hidden size might not achieve better performance on ROUGE and RMSE metrics. Thus, we set the hidden size of GRU to 512 in all experiments. We fix the decoder (encoder) to 2 layers when adjust the number of GRU layers of the encoder (decoder). Fig. 4 shows that our model performs best when the number of GRU layers for both encoder and decoder is set to 2.

Furthermore, recall that we only preserve reviews between active users and popular products. To evaluate the effect of K on the quality of summary generation and rating prediction, we further conduct an extra parameter analysis experiment. As shown in Fig. 5, we can see that our model performs relatively well on these two tasks when K is set to 10. When the K is larger, the computation complexity would increase. When users and products have less history reviews (e.g., $K = 5$), it is hard

Review: I gave this to my niece's son, although i was almost tempted to keep it myself... || ...||It could serve as a nightlight for your own little caped crusader. Of course, you get batman and robin figures to play with, as well as a batcycle, a bat-glider, and even a "bat claw" to make repairs to the batmobile. The features include a rotating turnstile. any batman fan, regardless of age, should have a great time with this product.

Predicted Rating:	4.31
Gold Rating:	5.0
Generated Summary:	a great toy for a little caped fan
Gold Summary:	great bat-toy for the batman fan

Review: I'm an unashamed adult collector of lego toys. while the ninjago theme didn't particularly appeal to me.||...|| However, if you aren't interested at all in the theme or just by looking at the pictures of this set, this set is worth passing up.in summary, great for younger kids! They will have blast. Older kids and adults, pass this one up.||...

Predicted Rating:	4.65
Gold Rating:	5.0
Generated Summary:	great for youger kids
Gold Summary:	great for the kids; adult fans, probably not

Fig. 7. Two cases in the *Toys* dataset.

to directly learn comprehensive representations for users and products from the review text.

Finally, we evaluate our model under different dropout ranging in (0,0.7]. The experimental results in Fig. 6 show that our model achieves better performance on both summary generation and rating prediction when dropout is set to 0.1.

5.4. Case study

To indicate the effect of our model more intuitively, we show two cases in Fig. 7, in which reviews are from the dataset "Toys". Here, we only select the important sentences since the reviews are too long. The results show that the generated summary reflects the user preference to product and the text is consistent with the user writing style. In the first case, our generated summary contains the keyword "caped" which is the important characteristic of the product "batman". In the second case, we can see that our generated summary and gold summary are both related to the important content marked with the color red. In total, our model captures the important content by incorporating the user preference embedding and the product characteristic embedding to the decoder attention mechanism.

This paper mainly addresses the problem that generates summary for the input review by leveraging rating information to control the sentiment tendency of the generated summary. In Table 5, we list some real cases generated by our model and the corresponding reference summaries. The first line of each case is the generated summary and the corresponding predicted rating, the second line is the corresponding reference summary and rating. The results show that both generated summaries and reference summaries mention the important product characteristic, such as "rice cooker" in the first case and "great for the pet shop fans" in the second case. It demonstrates that user preferences and the product characteristics are helpful to identify the important words in the input review. In addition, the ratings for generated summaries and reference summaries are similar. It demonstrates that generated summaries not only contain the salient information of input reviews but also have the same sentiment with the reference summaries.

Table 5
Generated summaries for some reviews.

Rating	Summaries
4.08	**Great little rice cooker**
5	Good & simple rice cooker
4.42	**great for littlest pet shop fans**
5	big fun for littlest pet shop fans
4.55	**a great toy for a little caped fan**
4	great bat-toy for the batman fan
4.76	**great for the lego fans**
4	great for the kids adult fans probably not
4.24	**The swivel design is awesome on dog hair**
4	Badly Balanced, Amazing on dog hair!
4.58	**Works - but it is a pain to clean**
4	It works fine, but I had to work too.
3.53	**It is not bad but Only can use with small vegetable...**
3	Only can use with small vegetable

6. Conclusion

In this paper, we propose a Rating-boosted Abstractive Review Summarization with personalized generation (RARS). We design a review-level attention to effectively learn the user preferences and product characteristics from their history reviews. Especially, we propose a personalized decoder which calculates the saliency score of words based on the learned user and product features to guide the generation phase. A multi-task learning framework further boosts the performance of both review summarization task and rating prediction task. The experimental results on the two tasks all demonstrate the effectiveness of our model. The case studies also indicate that our method could generate more impressive summaries for reviews. In the future, we are going to explore the deep interaction between the history reviews/summaries and the input review, which is beneficial to identify the important information from the history text and further enhance the summary generation.

CRediT authorship contribution statement

Hongyan Xu: Conceptualization, Methodology, Writing - original draft. **Hongtao Liu:** Methodology, Writing - review & editing. **Wang Zhang:** Investigation, Visualization. **Pengfei Jiao:** Methodology, Writing - review & editing. **Wenjun Wang:** Resources, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (2018YFC0832100, 2020YFC0833303), the National Natural Science Foundation of China (61902278, 61902279) and China Postdoctoral Science Foundation (No. 2019M650048).

References

[1] J. Li, H. Li, C. Zong, Towards personalized review summarization via user-aware sequence network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 6690–6697.

[2] A. See, P.J. Liu, C.D. Manning, Get to the point: Summarization with pointer-generator networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1073–1083.

[3] S. Ma, X. Sun, J. Lin, X. Ren, A hierarchical end-to-end model for jointly improving text summarization and sentiment classification, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018, pp. 4251–4257.

[4] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, 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), 2014, pp. 1724–1734.

[5] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, M. Sun, A unified model for extractive and abstractive summarization using inconsistency loss, ACL (2018) 132–141.

[6] S. Gehrmann, Y. Deng, A.M. Rush, Bottom-up abstractive summarization, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4098–4109.

[7] X. Zhang, F. Wei, M. Zhou, 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, 2019, pp. 5059–5069.

[8] D. Wang, P. Liu, Y. Zheng, X. Qiu, X. Huang, Heterogeneous graph neural networks for extractive document summarization, 2020, arXiv preprint arXiv:2004.12393.

[9] Y. Du, Q. Li, L. Wang, Y. He, Biomedical-domain pre-trained language model for extractive summarization, Knowl. Based Syst. 199 (2020) 105964.

[10] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 168–177.

[11] K. Ganesan, C. Zhai, J. Han, Opinosis: A graph based approach to abstractive summarization of highly redundant opinions, in: Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), 2010, pp. 340–348.

[12] W. Xiong, D. Litman, 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, 2014, pp. 1985–1995.

[13] G. Carenini, J.C.K. Cheung, A. Pauls, Multi-document summarization of evaluative text, Comput. Intell. 29 (4) (2013) 545–576.

[14] G. Di Fabbrizio, A. Stent, R. Gaizauskas, A hybrid approach to multi-document summarization of opinions in reviews, in: Proceedings of the 8th International Natural Language Generation Conference (INLG), 2014, pp. 54–63.

[15] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[16] Y. Lu, R. Dong, B. Smyth, Why I like it: multi-task learning for recommendation and explanation, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 4–12.

[17] L. Dong, S. Huang, F. Wei, M. Lapata, M. Zhou, K. Xu, 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, 2017, pp. 623–632.

[18] S. Gerani, Y. Mehdad, G. Carenini, R. Ng, B. Nejat, Abstractive summarization of product reviews using discourse structure, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1602–1613.

[19] L. Wang, W. Ling, Neural network-based abstract generation for opinions and arguments, NAACL (2016) 47–57.

[20] M. Yang, Q. Qu, Y. Shen, Q. Liu, W. Zhao, J. Zhu, Aspect and sentiment aware abstractive review summarization, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1110–1120.

[21] P. Li, Z. Wang, Z. Ren, L. Bing, W. Lam, Neural rating regression with abstractive tips generation for recommendation, in: SIGIR, ACM, 2017, pp. 345–354.

[22] J. Li, X. Wang, D. Yin, C. Zong, 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), 2019, pp. 2991–3001.

[23] H. Liu, X. Wan, Neural review summarization leveraging user and product information, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2389–2392.

[24] P. Li, Z. Wang, L. Bing, W. Lam, Persona-aware tips generation, in: The World Wide Web Conference, ACM, 2019, pp. 1006–1016.

[25] H.P. Chan, W. Chen, I. King, A unified dual-view model for review summarization and sentiment classification with inconsistency loss, in: Proceedings of the 43th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1191–1200.

[26] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.

[27] A. Mnih, R. Salakhutdinov, Probabilistic matrix factorization, in: NIPS, 2007, pp. 1257–1264.

[28] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: NIPS, 2001, pp. 556–562.

[29] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: KDD, ACM, 2008, pp. 426–434.

[30] J. McAuley, J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: RecSys, ACM, 2013, pp. 165–172.

[31] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: KDD, ACM, 2011, pp. 448–456.

[32] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, WWW (2018) 1583–1592.

[33] H. Liu, Y. Wang, Q. Peng, F. Wu, L. Gan, L. Pan, P. Jiao, Hybrid neural recommendation with joint deep representation learning of ratings and reviews, Neurocomputing 374 (2020) 77–85.

[34] L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, 2017, pp. 425–434.

[35] G. Pang, X. Wang, F. Hao, J. Xie, X. Wang, Y. Lin, X. Qin, ACNN-FM: a novel recommender with attention-based convolutional neural network and factorization machines, Knowl. Based Syst. 181 (2019).

[36] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, L. Redondo-Expósito, Automatic construction of multi-faceted user profiles using text clustering and its application to expert recommendation and filtering problems, Knowl. Based Syst. 190 (2020) 105337.

[37] R.C. Bagher, H. Hassanpour, H. Mashayekhi, User preferences modeling using dirichlet process mixture model for a content-based recommender system, Knowl. Based Syst. 163 (2019) 644–655.

[38] T. Pradhan, S. Pal, CNAVER: a content and network-based academic venue recommender system, Knowl. Based Syst. 189 (2020).

[39] J. Zhao, X. Geng, J. Zhou, Q. Sun, Y. Xiao, Z. Zhang, Z. Fu, Attribute mapping and autoencoder neural network based matrix factorization initialization for recommendation systems, Knowl. Based Syst. 166 (2019) 132–139.

[40] G. Erkan, D.R. Radev, Lexrank: Graph-based lexical centrality as salience in text summarization, JAIR 22 (2004) 457–479.

[41] M. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, 2015, arXiv: Computation and Language.

[42] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[43] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[44] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, 2004, pp. 74–81.

[45] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.