

A Unified Dual-view Model for Review Summarization and Sentiment Classification with Inconsistency Loss

Hou Pong Chan*
hpchan@cse.cuhk.edu.hk
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong

Wang Chen*
wchen@cse.cuhk.edu.hk
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong

Irwin King
king@cse.cuhk.edu.hk
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong

ABSTRACT

Acquiring accurate summarization and sentiment from user reviews is an essential component of modern e-commerce platforms. Review summarization aims at generating a concise summary that describes the key opinions and sentiment of a review, while sentiment classification aims to predict a sentiment label indicating the sentiment attitude of a review. To effectively leverage the shared sentiment information in both review summarization and sentiment classification tasks, we propose a novel dual-view model that jointly improves the performance of these two tasks. In our model, an encoder first learns a context representation for the review, then a summary decoder generates a review summary word by word. After that, a source-view sentiment classifier uses the encoded context representation to predict a sentiment label for the review, while a summary-view sentiment classifier uses the decoder hidden states to predict a sentiment label for the generated summary. During training, we introduce an inconsistency loss to penalize the disagreement between these two classifiers. It helps the decoder to generate a summary to have a consistent sentiment tendency with the review and also helps the two sentiment classifiers learn from each other. Experiment results on four real-world datasets from different domains demonstrate the effectiveness of our model.

CCS CONCEPTS

• **Information systems** → **Summarization; Sentiment analysis; Recommender systems.**

KEYWORDS

Review Summarization; Sentiment Classification; Multi-task Learning; Text Summarization

ACM Reference Format:

Hou Pong Chan, Wang Chen, and Irwin King. 2020. A Unified Dual-view Model for Review Summarization and Sentiment Classification with Inconsistency Loss. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, July 25–30, 2020, Virtual

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '20, July 25–30, 2020, Virtual Event, China

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-8016-4/20/07...\$15.00

<https://doi.org/10.1145/3397271.3401039>

Review: very colorful and decent for prep and serving small kids. bright colors are nice and food colors don't seem to transfer to the bowls. we use them most frequently for salad dressing when the kids have baby carrots or celery sticks. my complaint is that the bases are too small and consequently they tip over very easily. that's ok for ranch dressing or ketchup, not so good for soy sauce. if you flip them inside out it gives them a wider base which makes them more stable. since they flex they make their own funnel ... we have some stainless mini bowls which are almost an inch larger in diameter (but shallower) which makes them a little more handy ...

Summary: very colorful and handy, but tippy and smaller than expected.

Sentiment label: 3 out of 5

Figure 1: An example of truncated review and its corresponding summary and sentiment label.

Event, China. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3397271.3401039>

1 INTRODUCTION

User reviews are precious for companies to improve the quality of their services or products since reviews express the opinions of their customers. Nowadays, a company can easily collect reviews from users via e-commerce platforms and recommender systems, but it is difficult to read through all the wordy user reviews. Therefore, distilling salient information from user reviews is necessary. To achieve such a goal, review summarization and sentiment classification are continuously explored by plenty of researchers. The objective of review summarization is to generate a short and concise summary that expresses the key opinions and sentiment of the review. Sentiment classification is the task of predicting the sentiment label which indicates the sentiment attitude of the review. For example, a sentiment label ranges from 1 to 5, where 1 indicates the most negative attitude and 5 indicates the most positive attitude.

Figure 1 shows an example of a review with its summary and sentiment label. As shown in the summary, the user thinks the product is “colorful and handy, but tippy and smaller than expected”. The sentiment label of the review is 3 out of 5, indicating the review has a neutral sentiment attitude. We can observe that the sentiment tendency of the review summary is consistent with the sentiment label of the review, which indicates there exists a close tie between these two tasks [24]. Hence, they can benefit each other. Specifically, the sentiment classification task can guide a summarization model to capture the sentiment attitude of the review. Meanwhile, the review summarization task can remove the less-informative information

from the review, which can assist a sentiment classification model to predict the sentiment label of a review.

However, most existing methods can only solve either review summarization or sentiment classification problem. For review summarization, traditional extractive methods select important phrases or sentences from the input review text to produce an output summary. Recent abstractive methods for review summarization adopt the attentional encoder-decoder framework [1] to generate an output summary word by word, which can generate novel words that do not appear in the review and produce a more concise summary. For sentiment classification, early methods are based on support vector machines or statistical models, while recent methods employ recurrent neural network (RNN) [15]. Though some previous methods [16, 25] can predict both the sentiment label and the summary for a social media text, the sentiment classification and summarization modules are trained separately and they rely on rich hand-crafted features.

Recently, Ma et al. [24] proposed an end-to-end model that jointly improves the performance of review summarization and sentiment classification. In their model, an encoder first learns a context representation for the review, which captures the context and sentiment information. Based on this representation, the decoder iteratively computes a hidden state and uses it to generate a summary word from a predefined vocabulary until the end-of-sequence (EOS) token is generated. Meanwhile, an attention layer utilizes the decoder hidden states to attend the review to compute a weighted sum of the review context representation, which acts as a summary-aware context representation of the review. Next, the review context representation and the summary-aware context representation are fed to a max-pooling based classifier to predict the sentiment label. However, both of these representations collect sentiment information from the review only. Thus, the model does not fully utilize the sentiment information existing in the summary.

To effectively leverage the sentiment information in the review and the summary, we propose a dual-view model for joint review summarization and sentiment classification. In our model, the sentiment information in the review and the summary are modeled by the source-view and summary-view sentiment classifiers respectively. The source-view sentiment classifier uses the review context representation from the encoder to predict a sentiment label for the review, while the summary-view sentiment classifier utilizes the decoder hidden states to predict a sentiment label for the generated summary. The ground-truth sentiment label of the review will be used to compute a classification loss for both of these classifiers. In addition, we also introduce an inconsistency loss function to penalize the disagreement between these two classifiers.

By promoting the consistency between the source-view and summary-view classifiers, we encourage the sentiment information in the decoder states to be close to that in the review context representation, which helps the decoder to generate a summary that has the same sentiment attitude as the review. Moreover, the source-view and summary-view sentiment classifiers can learn from each other to improve the sentiment classification performance. This shares a similar spirit with the multi-view learning paradigm in semi-supervised learning [41]. Our model, therefore, provides more supervision signals for both review summarization and sentiment classification without additional training data. During testing, we

use the sentiment label predicted by the source-view classifier as the final classification output since the summary-view sentiment classifier will be affected by the exposure bias issue [33] in the testing stage (more details in Section 4.4).

We conduct extensive experiments to evaluate the performance of our dual-view model. Experiment results on four real-world datasets from different domains demonstrate that our model outperforms strong baselines on both review summarization and sentiment classification. The ablation study shows the effectiveness of each individual component of our model. Furthermore, we also compare the classification results of our source-view sentiment classifier, summary-view sentiment classifier, and a merged sentiment classifier that combines the former two classifiers.

We summarize our main contributions as follows: (1) we propose a novel dual-view model for jointly improving the performance of review summarization and sentiment classification; (2) we introduce an inconsistency loss to penalize the inconsistency between our source-view and summary-view sentiment classifiers, which benefits both review summarization and sentiment classification; and (3) experimental results on benchmark datasets show that our model outperforms the state-of-the-art models for the joint review summarization and sentiment classification task.

2 RELATED WORK

Opinion Summarization. Review summarization belongs to the area of opinion summarization [8, 9, 12, 21, 29, 36]. Early approaches for opinion summarization are extractive, i.e., they can only produce words that appear in the input document. Ganesan et al. [9] proposed a graph-based approach. It first converts the input opinion text into a directed graph. Then it applies heuristic rules to score different paths that encode valid sentences and takes the top-ranked paths as the output summary. Hu and Liu [18] proposed a method that first identifies product features mentioned in the reviews and then extracts opinion sentences for the identified features. An unsupervised learning method is proposed to extract a review summary by exploiting the helpfulness scores in reviews [40].

On the other hand, abstractive approaches for opinion summarization can generate novel words that do not appear in the input document. Gerani et al. [12] proposed a template filling strategy to generate a review summary. Wang and Ling [39] applied the attentional encoder-decoder model to generate an abstractive summary for opinionated documents. All of the above methods consider opinion summarization in multiple documents setting, while our work considers opinion summarization on single document setting. Li et al. [21] studied the problem of personalized review summarization on single review setting. They incorporated a user’s frequently used words into the encoder-decoder model to generate a user-aware summary. In contrast, we focus on modeling the shared sentiment information between the tasks of review summarization and sentiment classification, which is orthogonal to the personalized review generation problem. Hsu et al. [17] proposed a unified model for extractive and abstractive summarization with an inconsistency loss to penalize the disagreement between the extractive and abstractive attention scores. Compared to their model, which penalizes the inconsistencies between two different attention methods on the same view (i.e., the source input), we introduce an inconsistency

loss to penalize the outputs of two different classifiers on different views (i.e., the source view and the summary view).

Review Sentiment Classification. Review sentiment classification aims at analyzing online consumer reviews and predicting the sentiment attitude of a consumer towards a product. The review sentiment classification tasks can be categorized into three groups: document-level, sentence-level, and aspect-level sentiment classification [6, 45]. We focus on document-level sentiment classification that is to assign an overall sentiment orientation to the input document [45], which is usually treated as a kind of document classification task [30, 31]. Traditional methods focus on designing effective features that are used in either supervised learning methods [10, 32] or graph-based semi-supervised methods [13]. Recently, neural network based methods which do not require hand-crafted features achieve state-of-the-art performance on this task. For example, Tang et al. [35] proposed a neural network based hierarchical encoding process to learn an effective review representation. Hierarchical attention mechanisms [43, 44, 47] are also extensively explored in this task for constructing an effective representation of the review. Different from previous methods that are designed only for the review sentiment classification task, we propose a unified model for simultaneously generating the summary of the review and classifying the review sentiment.

Joint Text Summarization and Classification. There are different joint models for both text summarization and classification. Cao et al. [4] proposed a neural network model that jointly classifies the category and extracts summary sentences for a group of news articles, but it can only improve the performance of text summarization. Yang et al. [42] proposed a joint model that uses domain classification as an auxiliary task to improve the performance of review summarization. Moreover, their model uses different lexicons to find out sentiment words and aspect words from the review text, and then incorporates them into the decoder via attention mechanism to generate aspect/sentiment-aware review summaries. The above two methods use a domain classification task to improve the performance of summarization, while our method jointly improves the performance of both review summarization and sentiment classification. Two models [16, 25] were proposed to jointly extract a summary and predict the sentiment label for a social media post, but the summarization module and classification module of these models are trained separately and they require rich hand-crafted features. Recently, Ma et al. [24] proposed an end-to-end neural model that jointly improves the performance of the review summarization and sentiment classification tasks. However, their sentiment classifier only collects sentiment information from the review. Our dual-view model has a source-view sentiment classifier and a summary-view sentiment classifier to model the sentiment information in the review and the summary respectively. We also introduce an inconsistency loss to encourage the consistency between these two classifiers.

3 PRELIMINARY

Notations. We use bold lowercase characters to denote vectors, bold upper case characters to denote matrices and calligraphy characters to denote sets. We use \mathbf{W} and \mathbf{b} to denote a projection matrix and a bias vector in a neural network layer.

Problem definition. We formally define the problem of review summarization and sentiment classification as follows. Given a review text \mathbf{x} , we output the summary \mathbf{y} and sentiment label z of the review text. The review text \mathbf{x} and its summary \mathbf{y} are sequences of words, i.e., $\mathbf{x} = [x_1, \dots, x_{L_x}]$ and $\mathbf{y} = [y_1, \dots, y_{L_y}]$, where L_x and L_y denotes the numbers of word in \mathbf{x} and \mathbf{y} respectively. The sentiment label $z \in \{1, 2, \dots, K\}$ is an integer that indicates the sentiment attitude of the review text, where 1 denotes the most negative sentiment and K denotes the most positive sentiment.

4 MODEL ARCHITECTURE

4.1 Overview

Our joint model consists of three major modules: (M1) shared text encoder module, (M2) summary decoder module, and (M3) dual-view sentiment classification module. The input review text \mathbf{x} is first encoded by the shared text encoder into context-aware representations $\tilde{\mathbf{H}} = [\tilde{\mathbf{h}}_1, \dots, \tilde{\mathbf{h}}_{L_x}]$, which forms a memory bank for the summary decoder module and the dual-view sentiment classification module. Then, the summary decoder uses the memory bank from the encoder to compute a sequence of hidden states $\mathbf{S} = [s_1, \dots, s_{L_y}]$ and generates a review summary word by word. The ground-truth summary is used to compute a summary generation loss for the model. Our dual-view classification module consists of a source-view sentiment classifier and a summary-view sentiment classifier. The source-view sentiment classifier reads the encoder memory bank $\tilde{\mathbf{H}}$ and predicts a sentiment label z for the review, while the summary-view sentiment classifier uses the decoder hidden states \mathbf{S} to predict a sentiment label z' for the generated summary. We use the ground-truth sentiment label of the review to compute a sentiment classification loss for both of these sentiment classifiers. Besides, we also introduce an inconsistency loss function to penalize the disagreement between these two classifiers. We jointly minimize all the above loss functions by a multi-task learning framework. During testing, we use the sentiment label predicted by the source-view classifier as the output sentiment label. Figure 2 illustrates the overall architecture of our model. We describe the details of each module as follows.

4.2 Shared Text Encoder (M1)

The shared text encoder converts the review text into a memory bank for the sentiment classification and summary decoder modules. First, the encoder reads an input sequence $\mathbf{x} = (x_1, \dots, x_{L_x})$ and uses a lookup table to convert each input word x_i to a word embedding vector $\mathbf{x}_i \in \mathbb{R}^{d_e}$. To incorporate the contextual information of the review text into the representation of each word, we feed each embedding vector \mathbf{x}_i to a bi-directional Gated-Recurrent Unit (GRU) [7] to learn a shallow hidden representation $\mathbf{u}_i \in \mathbb{R}^d$. More specifically, a bi-directional GRU consists of a forward GRU that reads the embedding sequence from \mathbf{x}_1 to \mathbf{x}_{L_x} and a backward GRU that reads from \mathbf{x}_{L_x} to \mathbf{x}_1 :

$$\vec{\mathbf{u}}_i = \text{GRU}_{11}(\mathbf{x}_i, \vec{\mathbf{u}}_{i-1}), \quad (1)$$

$$\overleftarrow{\mathbf{u}}_i = \text{GRU}_{12}(\mathbf{x}_i, \overleftarrow{\mathbf{u}}_{i+1}), \quad (2)$$

where $\vec{\mathbf{u}}_i \in \mathbb{R}^{d/2}$ and $\overleftarrow{\mathbf{u}}_i \in \mathbb{R}^{d/2}$ denotes the hidden states of the forward GRU and backward GRU respectively. We concatenate

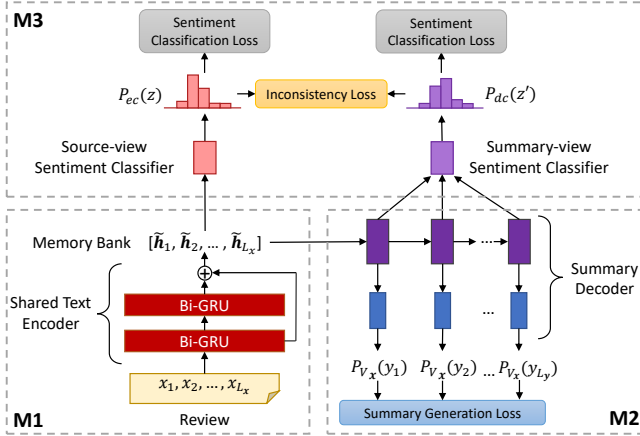


Figure 2: Overall model architecture consisting of (M1) shared text encoder, (M2) summary decoder, and (M3) dual-view sentiment classification module. The shared text encoder converts the input review text into a memory bank. Based on the memory bank, the summary decoder generates the review summary word by word and receives a summary generation loss. The source-view (summary-view) sentiment classifier uses the memory bank (hidden states) from the encoder (decoder) to predict a sentiment label for the review (summary) and it receives a sentiment classification loss. An inconsistency loss is applied to penalize the disagreement between the source-view and summary-view sentiment classifiers.

\vec{u}_i and \overleftarrow{u}_i to form the shallow hidden representation for x_i , i.e., $u_i = [\vec{u}_i; \overleftarrow{u}_i]$.

Next, we pass the shallow hidden representations to another bi-directional GRU to model more complex interactions among the words in the review text:

$$\vec{h}_i = \text{GRU}_{21}(u_i, \vec{h}_{t-1}), \quad (3)$$

$$\overleftarrow{h}_i = \text{GRU}_{22}(u_i, \overleftarrow{h}_{t+1}), \quad (4)$$

where $\vec{h}_i \in \mathbb{R}^{d/2}$ and $\overleftarrow{h}_i \in \mathbb{R}^{d/2}$. We concatenate $\vec{h}_i \in \mathbb{R}^{d/2}$ and \overleftarrow{h}_i to form $h_i \in \mathbb{R}^d$. Then we apply a residual connection from the shallow hidden representation u_i to h_i , which is standard technique to avoid gradient vanishing problem in deep neural networks [14], as shown in the following equation:

$$\tilde{h}_i = \lambda h_i + (1 - \lambda) u_i, \quad (5)$$

where $\lambda \in [0, 1]$ is a hyperparameter. We regard the final encoder representations $[\tilde{h}_1, \dots, \tilde{h}_{L_x}]$ as the memory bank for the later summary decoder and dual-view classification modules.

4.3 Summary Decoder (M2)

The decoder uses a forward GRU to generate an output summary $y = [y_1, \dots, y_t, \dots, y_{L_y}]$ step by step. The architecture of our summary decoder follows the decoder of the pointer generator network [34]. At each decoding step t , the forward GRU reads the

embedding of the previous prediction y_{t-1} and the previous decoder hidden state s_{t-1} to yield the current decoder hidden state:

$$s_t = \text{GRU}_3(y_{t-1}, s_{t-1}), \quad (6)$$

where $s_t \in \mathbb{R}^d$, y_0 is the embedding of the start token. To gather relevant information from the document, a neural attention layer [1] is then applied to compute an attention score $a_{t,i}$ between the current decoder hidden state s_t and each of the vectors in the encoder memory bank \tilde{h}_i :

$$\alpha_{t,i} = v^T \tanh(W_h \tilde{h}_i + W_s s_t + \mathbf{b}_{attn}), \quad (7)$$

$$a_{t,i} = \frac{\exp(\alpha_{t,i})}{\sum_{j=1}^{L_x} \exp(\alpha_{t,j})}, \quad (8)$$

where $v \in \mathbb{R}^{d'}$, $W_h \in \mathbb{R}^{d' \times d}$, $W_s \in \mathbb{R}^{d' \times d}$ and $\mathbf{b}_{attn} \in \mathbb{R}^{d'}$ are model parameters. The attention score is next used to compute a weighted sum of the memory bank vectors and produce an aggregated vector \tilde{h}_t^* , which acts as the representation of the input sequence x at time t : $\tilde{h}_t^* = \sum_{i=1}^{L_x} a_{t,i} \tilde{h}_i$. After that, we use \tilde{h}_t^* and the decoder hidden state s_t to compute a probability distribution over the words in a predefined vocabulary \mathcal{V} , as shown in the following equation,

$$P_{\mathcal{V}}(y_t | y_{1:t-1}, x) = \text{softmax}(W_V(W_V [s_t; \tilde{h}_t^*] + \mathbf{b}_{V'}) + \mathbf{b}_V), \quad (9)$$

where $y_{1:t-1}$ denotes the partial sequence of previous generated words, $[y_1, \dots, y_{t-1}]$, and $W_V \in \mathbb{R}^{|\mathcal{V}| \times d}$, $W_{V'} \in \mathbb{R}^{d \times d}$, $\mathbf{b}_V \in \mathbb{R}^{|\mathcal{V}|}$, $\mathbf{b}_{V'} \in \mathbb{R}^d$ are trainable parameters.

Although we can directly use $P_{\mathcal{V}}$ as our final prediction distribution, the decoder cannot generate out-of-vocabulary (OOV) words. To address this problem, we adopt the copy mechanism from [34] to empower the decoder to predict OOV words by directly copying words from the input review. In the copy mechanism, we first compute a soft gate $p_{gen} \in [0, 1]$ between generating a word from the predefined vocabulary \mathcal{V} according to $P_{\mathcal{V}}$ and copying a word from the source text x according to the attention distribution:

$$p_{gen} = \sigma(v_g^T [\tilde{h}_t^*; s_t; y_{t-1}] + \mathbf{b}_{gen}), \quad (10)$$

where $v_g \in \mathbb{R}^{2d+d_e}$ and $\mathbf{b}_{gen} \in \mathbb{R}$ are trainable parameters and $\sigma(\cdot)$ denotes the logistic sigmoid function. Next, we define a dynamic vocabulary \mathcal{V}_x as the union of \mathcal{V} and all the words appear in the review x . Finally, we can compute a probability distribution $P_{\mathcal{V}_x}$ over the words in the dynamic vocabulary as follows:

$$P_{\mathcal{V}_x}(y_t) = p_{gen} P_{\mathcal{V}}(y_t) + (1 - p_{gen}) \sum_{i:x_i=y_t} a_{t,i}, \quad (11)$$

where we use $P_{\mathcal{V}_x}(y_t)$ to denote $P_{\mathcal{V}_x}(y_t | y_{1:t-1}, x)$ and $P_{\mathcal{V}}(y_t)$ to denote $P_{\mathcal{V}}(y_t | y_{1:t-1}, x)$ for brevity. In Eq. (11), we define $P_{\mathcal{V}}(y_t) = 0$ for all $y_t \notin \mathcal{V}$ (OOV words). If y_t does not appear in x , the copying probability, $\sum_{i:x_i=y_t} a_{t,i}$, will be zero.

Summary generation loss function. We use the negative log-likelihood of the ground-truth summary y^* as the loss function for the review summarization task:

$$\mathcal{L}_{gen} = - \sum_{t=1}^{L_{y^*}} \log P_{\mathcal{V}_x}(y_t^* | y_{1:t-1}^*, x), \quad (12)$$

where L_{y^*} denotes the number of words in the ground-truth review summary y^* .

Inference. In the testing stage, we use beam search to generate the output summary from the summary decoder. This is a standard technique to approximate the output sequence that achieves the highest generation probability.

4.4 Dual-view Sentiment Classification Module (M3)

We propose a dual-view sentiment classification module to learn a sentiment label for the review. It consists of a source-view sentiment classifier and a summary-view sentiment classifier.

Source-view sentiment classifier. The source-view sentiment classifier utilizes the encoder memory bank $\tilde{\mathbf{H}} = [\tilde{\mathbf{h}}_1, \dots, \tilde{\mathbf{h}}_{L_x}]$ to predict a sentiment label for the review. Since not all words in the review contribute equally to the prediction of sentiment label, we apply the attention mechanism [1] to aggregate sentiment information from the encoder memory bank into a sentiment context vector. An additional glimpse operation [38] is incorporated in this aggregation process since the glimpse operation has been shown that it can improve the performance of several attention-based classification models [2, 5].

First, a trainable query vector $q \in \mathbb{R}^{d_q}$ attends the encoder memory bank and produce a glimpse vector $\mathbf{g} \in \mathbb{R}^d$ as follows:

$$\alpha_i^g = \mathbf{v}_g^T \tanh(\mathbf{W}_{gh}\tilde{\mathbf{h}}_i + \mathbf{W}_{gq}\mathbf{q} + \mathbf{b}_g), \quad (13)$$

$$a_i^g = \frac{\exp(\alpha_i^g)}{\sum_{j=1}^{L_x} \exp(\alpha_j^g)}, \quad (14)$$

$$\mathbf{g} = \sum_{i=1}^{L_x} a_i^g \tilde{\mathbf{h}}_i, \quad (15)$$

where $\mathbf{W}_{gh} \in \mathbb{R}^{d' \times d}$, $\mathbf{W}_{gq} \in \mathbb{R}^{d' \times d}$, $\mathbf{b}_g \in \mathbb{R}^{d'}$, $\mathbf{v}_g \in \mathbb{R}^{d'}$ are trainable model parameters.

Then, the glimpse vector \mathbf{g} attends the memory bank again to compute a review sentiment context vector $\mathbf{e} \in \mathbb{R}^d$: $\mathbf{e} = \sum_{i=1}^{L_x} a_i^e \tilde{\mathbf{h}}_i$, where a_i^e is the corresponding attention weight.

The classifier is a two-layer feed-forward neural network using a rectified linear unit (ReLU) as the activation function [24]. The softmax function is applied at the output of the network to produce a probability distribution over the sentiment label:

$$P_{ec}(z|\mathbf{x}) = \text{softmax}(\mathbf{W}_{z2}(\text{ReLU}(\mathbf{W}_{z1}\mathbf{e} + \mathbf{b}_{z1})) + \mathbf{b}_{z2}), \quad (16)$$

where $\mathbf{W}_{z1} \in \mathbb{R}^{d_z \times d}$, $\mathbf{W}_{z2} \in \mathbb{R}^{K \times d_z}$, $\mathbf{b}_{z1} \in \mathbb{R}^{d_z}$, $\mathbf{b}_{z2} \in \mathbb{R}^K$ are model parameters. The sentiment label with the highest probability is treated as the predicted sentiment label for the review.

Summary-view sentiment classifier. The summary-view sentiment classifier uses the decoder hidden states $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_{L_y}]$ to predict a sentiment label for the generated summary. Since each decoder hidden state is used to generate a summary word, we treat the decoder hidden states as the representation for the generated summary. Then, we apply attention mechanism [1] with glimpse operation [38] to compute a sentiment context vector \mathbf{e}' for the generated summary. The architecture of the summary-view sentiment classifier is the same as the source-view sentiment classifier but with another set of parameters. The only difference is that it uses the decoder hidden states as the input instead of the encoder memory bank, i.e., all the $\tilde{\mathbf{h}}_i$ terms in the equations of the source-view

sentiment classifier are replaced by \mathbf{s}_i . Then, the summary-view sentiment classifier outputs a probability distribution over the sentiment label for the generated summary: $P_{dc}(z'|\mathbf{x}, \mathbf{y})$.

Sentiment classification loss function. We use negative log-likelihood as the classification loss function for both the source-view sentiment classifier and the summary-view sentiment classifier:

$$\mathcal{L}_{ec} = -\log P_{ec}(z^*|\mathbf{x}), \quad (17)$$

$$\mathcal{L}_{dc} = -\log P_{dc}(z^*|\mathbf{x}, \mathbf{y}), \quad (18)$$

where z^* denotes the ground-truth sentiment label, \mathcal{L}_{ec} and \mathcal{L}_{dc} denote the classification loss for the source-view and the summary-view sentiment classifiers respectively.

Inconsistency loss function. We introduce an inconsistency loss to penalize the disagreement between the source-view sentiment classifier and the summary-view sentiment classifier. The intuition is that the review summary should have the same sentiment attitude as the input review. We define our inconsistency loss function as the Kullback-Leibler (KL) divergence between P_{ec} and P_{dc} :

$$\mathcal{L}_{inc} = D_{KL}(P_{ec}||P_{dc}) = \sum_{k=1}^K P_{ec}(k|\mathbf{x}) \log \frac{P_{ec}(k|\mathbf{x})}{P_{dc}(k|\mathbf{x}, \mathbf{y})}. \quad (19)$$

Since the summary-view sentiment classifier uses the decoder hidden states to predict the sentiment label for the generated summary, the inconsistency loss encourages the sentiment information in the decoder states to be close to that in the encoder memory bank. Thus, it helps the decoder to generate a summary that has a consistent sentiment with the review. Moreover, the source-view and summary-view sentiment classifiers can learn from each other to improve the sentiment classification performance.

Inference. In the testing stage, we use the sentiment label predicted by the source-view classifier as the final classification prediction. The reason is that the decoder suffers from the well-known exposure bias problem [33] during testing, which affects the performance of the summary-view classifier when inference. We conduct experiments to analyze the influence of the exposure bias on the classification performance and provide more discussions in Section 6.3.

4.5 Multi-task Training Objective

We adopt a multi-task learning framework to jointly minimize the review summarization loss, source-view sentiment classification loss, summary-view sentiment classification loss, and inconsistency loss. The objective function is shown as follows.

$$\mathcal{L} = \gamma_1 \mathcal{L}_{gen} + \gamma_2 \mathcal{L}_{ec} + \gamma_3 \mathcal{L}_{dc} + \gamma_4 \mathcal{L}_{inc}, \quad (20)$$

where $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ are hyper-parameters that controls the weights of these four loss functions. We set $\gamma_1 = 0.8, \gamma_2 = 0.1, \gamma_3 = 0.1, \gamma_4 = 0.1$ after fine-tuning on the validation datasets. Thus, each module of our joint model can be trained end-to-end.

5 EXPERIMENTAL SETUP

5.1 Datasets

In this work, we conduct experiments on four real-world datasets from different domains. The datasets are collected from the Amazon 5-core review repository [27]. We adopt product reviews from the following four domains as our datasets: **Sports & Outdoors**;

Table 1: Statistics of the datasets. “Ave. RL” denotes the average review length (in words) in the training set. “Ave. SL” denotes the average summary length in the training set. “Copy Ratio” indicates the ratio of the summary words that are copied from the review in the training set. “Rating k ” means the ratio of the data samples with sentiment label k in the training set.

Dataset	Training	Validation	Testing	Ave. RL	Ave. SL	Copy Ratio	Rating 1	Rating 2	Rating 3	Rating 4	Rating 5
Sports	183,714	9,000	9,000	108.3	6.7	58.5%	3.3%	3.9%	9.1%	22.9%	60.8%
Movies	1,200,601	20,000	20,000	167.1	6.6	58.8%	6.3%	6.3%	12.6%	23.3%	51.5%
Toys	104,296	8,000	8,000	125.9	6.8	61.0%	2.9%	4.1%	11.0%	24.0%	58.0%
Home	367,395	10,000	10,000	120.9	6.8	59.8%	5.3%	4.9%	9.1%	20.1%	60.6%

Movies & TV; Toys & Games; and Home & Kitchen. In our experiments, each data sample consists of a review text, a summary, and a rating. We regard the rating as a sentiment label, which is an integer in the range of [1, 5]. For text preprocessing, we lowercase all the letters and tokenize the text using Stanford CoreNLP [26]. We append a period to a summary sentence if it is not ended properly. To reduce the noise in these datasets, we filter out data samples when the review length is less than 16 or longer than 800, or the summary length is less than 4. We randomly split each dataset into training, validation, and testing sets. The statistics of these datasets are shown in Table 1.

5.2 Evaluation Metrics

For review summarization, we use **ROUGE score** [23] as the evaluation metric, which is a standard evaluation metric in the field of text summarization [22, 34]. Following [24], we use ROUGE-1, ROUGE-2, and ROUGE-L scores to evaluate the qualities of the generated review summaries. ROUGE-1 and ROUGE-2 measure the overlapping uni-grams and bi-grams between the generated review summary y and the ground-truth review summary y^* . ROUGE-L measures the longest common subsequence between the generated summary and the ground-truth review summary, we refer the readers to [23] for details.

For sentiment classification, we use the **macro F_1 score** and the **balanced accuracy** [3] as the evaluation metrics. We denote the macro F_1 score as “M. F_1 ” and the balanced accuracy as “B. Acc”. From Table 1, we can observe that the class distribution of the sentiment labels is imbalanced. Therefore, we do not use overall accuracy as the evaluation metric. To compute macro F_1 score, we first calculate the precision p_i and recall r_i for each individual sentiment class i . Next, we compute the macro-averaged precision and recall as follows, $p_{macro} = \sum_{i=1}^K p_i / K$, $r_{macro} = \sum_{i=1}^K r_i / K$. The macro F_1 score is the harmonic mean of p_{macro} and r_{macro} . The balanced accuracy is a variant of the accuracy metric for imbalanced datasets [3, 19]. It is defined as the macro-average of the recall obtained on each class, i.e., balanced accuracy = r_{macro} .

5.3 Baselines

Our baselines are categorized into three groups: (1) summarization-only models; (2) sentiment-only models; and (3) joint models. We use the following summarization-only models as our review summarization baselines.

- **PGNet** [34]: A popular summarization model which is based on the encoder-decoder framework with attention and copy mechanisms.

- **C.Transformer** [11]: The CopyTransformer model that enhances the state-of-the-art Transformer model [37] with copy mechanism [34] for abstractive summarization.

The following sentiment-only model are employed as one of our sentiment classification baselines.

- **BiGRU+Attention**: A bi-directional GRU layer [7] first encodes the input review into hidden states. Then it uses an attention mechanism [1] with glimpse operation [38] to aggregate information from the encoder hidden states and produce a vector. The vector is then fed through a two-layer feed-forward neural network to predict the sentiment label.
- **DARLM**: [46] The state-of-the-art model for sentence classification.

We also use the following joint models as the baselines of both review summarization and sentiment classification.

- **HSSC** [24]: The state-of-the-art model for jointly improving review summarization and sentiment classification.
- **Max** [24]: A bi-directional GRU layer first encodes the input review into hidden states. These hidden states are then shared by a summary decoder and a sentiment classifier. The sentiment classifier uses max pooling to aggregate the the encoder hidden states into a vector, which is then fed through a two-layer feed-forward neural network to predict the sentiment label.
- **HSSC+copy**: We incorporate the copy mechanism [34] into the HSSC model [24] as a strong baseline.
- **Max+copy**: We also incorporate the copy mechanism [34] into the Max model as another strong baseline.

5.4 Implementation Details

We train a word2vec [28] with a dimension of 128 (i.e., $d_e = 128$) on the training set of each dataset to initialize the word embeddings of all the models including baselines. All the initialized embeddings are fine-tuned by back-propagation during training. The vocabulary \mathcal{V} is defined as the 50,000 most frequent words in the training set. The hidden sizes d , d' , d_q , and d_z are set to 512. The λ in the residual connection is set to 0.5. The initial hidden states of the GRU cells in shared text encoder are zero vectors, while the initial hidden states of the GRU cell in the summary decoder is set to $[\lambda \vec{h}_{L_x} + (1 - \lambda) \vec{u}_{L_x}; \lambda \vec{h}_1 + (1 - \lambda) \vec{u}_1]$. A dropout layer with $p = 0.1$ is applied at the two-layer feed-forward neural networks in source-view and summary-view sentiment classifiers. The hidden sizes of the GRU cells and the feed-forward neural networks in our baselines are set to 512, which is the same as our model. During training, we truncate the input review text to 400 tokens and the

output summary to 100 tokens. We use the Adam optimization algorithm [20] with a batch size of 32 and an initial learning rate of 0.001. We use the gradient clipping of 2.0 using L-2 norm. The learning rate is reduced by half if the validation loss stops dropping. We apply early stopping when the validation loss stops decreasing for three consecutive checkpoints. During testing, we use the beam search algorithm with a beam size of 5 and a maximum depth of 120 for all models. We repeat all experiments five times with different random seeds and the averaged results are reported¹.

6 RESULTS ANALYSIS

Our experiments are intended to address the following research questions.

- **RQ1:** What is the performance of our proposed model on review summarization and sentiment classification?
- **RQ2:** What is the impact of each component of our model on the overall performance?
- **RQ3:** Which of the source-view and summary-view classifiers is better? Can we further improve the sentiment classification performance if we combine the source-view and summary-view classifiers by ensemble?
- **RQ4:** Are the generated review summaries consistent with the predicted sentiment labels?

6.1 Main Results (RQ1)

We show the review summarization results on the four datasets in Table 2. We note that our dual-view model achieves the best performance on almost all the metrics among all the four real-world datasets, demonstrating the effectiveness of our model to summarize a product review on different domains. We also conduct a significance test comparing our model with HSSC, Max, and PGNet. The results show our dual-view model significantly outperforms these three baselines on most of the metrics (paired t-test, $p < 0.05$).

The review sentiment classification results are shown in Table 3. For our dual-view model, the classification results of the source-view classifier are reported in this table. We observe that our dual-view model consistently outperforms all the baseline methods on all the datasets. These results show that our model can predict more accurate sentiment labels than baselines. Besides, the significance test results comparing with HSSC and Max indicate the improvements by our dual-view model are significant on most of the metrics (paired t-test, $p < 0.05$).

6.2 Ablation Study (RQ2)

We conduct an ablation study to verify the effectiveness of each important component of our model. The results on the Sports dataset are displayed in Table 4. “-I” denotes that we do not incorporate the inconsistency loss when training. By comparing the performance of the full model and “-I” in the table, we observe that after removing the inconsistency loss, the performance of both review summarization and sentiment classification drops obviously. Our experiment results on the Sports validation set also show that our inconsistency loss substantially reduces the number of inconsistent

predicted sentiment labels between the source-view and summary-view sentiment classifiers from 11.9% to 6.3%. If we replace the attention mechanism in classifiers with a max-pooling operation (i.e., compare “Full” and “-A” in the table), the performance decreases as we anticipated. We also find that after removing the residual connection in the encoder (i.e., compare “Full” and “-R”), the performance of both review summarization and sentiment classification degrades, which suggests that the residual connection module is effective for both tasks. Moreover, we note that the copy mechanism (i.e., compare “Full” and “-C”) is helpful for both review summarization and sentiment classification.

6.3 Classifier Ensemble (RQ3)

Though Table 3 reports the performance of the source-view sentiment classifier of our dual-view model, we are also interested in the performance of the summary-view sentiment classifier of our model and whether merging these two classifiers can improve the sentiment classification performance. To combine the sentiment label predictions from the source-view and summary-view sentiment classifiers, we average their predicted sentiment label probability distributions into a merged prediction probability distribution: $p_{mrg} = \frac{P_{ec} + P_{dc}}{2}$. We report the macro F_1 scores of the source-view, summary view, and merged sentiment classifiers in Table 5.

From this table, we note that the source-view classifier achieves the best results on three datasets. Thus, combining the source-view and summary-view sentiment classifiers does not yield better performance in most of the cases. Moreover, we also find that the source-view sentiment classifier consistently outperforms the summary-view sentiment classifier on all of these datasets. The main reason is that the exposure bias issue [33] of RNN decoder affects the performance of the summary-view sentiment classifier during testing. More specifically, in the training stage, previous ground-truth summary tokens are fed into the decoder to predict the next summary token (i.e., teacher-forcing) and the hidden states of the decoder can be regarded as the hidden representations of the ground-truth summary. In the testing stage, we cannot access the ground-truth summary. The previously-predicted tokens are fed into the decoder to predict the next summary token and the errors are accumulated in the decoder hidden states. Therefore, the summary-view classifier, which is based on the decoder hidden states, has a worse performance during testing.

We conduct experiments to demonstrate the effects of the exposure bias problem. Table 6 displays the macro F_1 scores of different sentiment classifiers with and without teacher-forcing. It is observed that when teacher-forcing is applied (i.e., w/ TF in the table), the summary-view classifier outperforms the source-view classifier by a large margin. However, its performance drops dramatically on both validation and testing sets after removing teacher-forcing (i.e., w/o TF). As we anticipated, the source-view classifier is not affected by whether teacher-forcing is adopted since it performs sentiment classification from the source input view instead of the summary view. The results of balanced accuracy scores show similar observations and we do not report them in Table 5 and 6 for brevity. These results suggest a future work of alleviating the performance gap of the summary-view sentiment classifier.

¹Source code is available at https://github.com/kenchan0226/dual_view_review_sum

Table 2: Review summarization results on the four datasets. We use ‘R’ to denote recall and ‘P’ to denote precision.

Dataset	Method	ROUGE-1			ROUGE-2			ROUGE-L		
		R	P	F ₁	R	P	F ₁	R	P	F ₁
Sports	PGNet	14.78±.21	19.79±.22	16.13±.21	6.13±.14	8.20±.16	6.62±.15	14.58±.20	19.46±.22	15.89±.21
	C.Transformer	13.73±.11	18.46±.12	15.02±.11	5.13±.12	6.88±.16	5.56±.13	13.54±.11	18.13±.12	14.80±.11
	HSSC	14.44±.25	19.24±.30	15.74±.26	5.52±.13	7.24±.16	5.93±.13	14.24±.24	18.91±.30	15.50±.25
	Max	14.36±.29	19.20±.30	15.67±.27	5.46±.14	7.22±.17	5.88±.13	14.14±.28	18.85±.29	15.41±.26
	HSSC+copy	14.64±.29	19.61±.41	15.98±.32	5.95±.13	7.98±.12	6.43±.12	14.43±.28	19.26±.39	15.74±.31
	Max+copy	14.75±.29	19.86±.37	16.15±.32	6.11±.17	8.22±.23	6.62±.18	14.56±.27	19.53±.35	15.92±.30
	Dual-view	15.39±.22	20.53±.32	16.79±.26	6.46±.07	8.63±.13	6.98±.10	15.18±.22	20.19±.31	16.55±.25
Movie	PGNet	12.67±.05	17.76±.06	14.04±.05	5.14±.07	7.38±.06	5.66±.06	12.40±.4	17.32±.05	13.72±.4
	C.Transformer	12.09±.14	16.78±.14	13.34±.13	4.46±.05	6.30±.07	4.89±.05	11.81±.14	16.33±.14	13.01±.13
	HSSC	12.59±.06	17.62±.09	13.89±.07	4.62±.05	6.59±.07	5.06±.05	12.27±.06	17.12±.09	13.53±.08
	Max	12.49±.10	17.53±.21	13.82±.14	4.70±.16	6.72±.34	5.16±.21	12.20±.10	17.07±.22	13.48±.15
	HSSC+copy	12.66±.06	17.92±.12	14.08±.05	5.06±.07	7.37±.09	5.60±.07	12.39±.06	17.47±.14	13.76±.07
	Max+copy	12.61±.04	17.81±.11	14.01±.06	5.04±.05	7.32±.12	5.57±.07	12.34±.04	17.38±.09	13.69±.05
	Dual-view	12.84±.07	17.98±.04	14.22±.06	5.22±.05	7.48±.06	5.75±.05	12.57±.07	17.55±.07	13.90±.06
Toys	PGNet	14.77±.24	20.54±.43	16.40±.27	6.18±.18	8.47±.19	6.74±.17	14.53±.23	20.13±.40	16.11±.25
	C.Transformer	12.57±.31	17.53±.53	13.94±.35	4.76±.08	6.42±.19	5.14±.10	12.36±.29	17.16±.51	13.69±.32
	HSSC	14.04±.20	19.29±.45	15.48±.26	5.24±.11	7.03±.16	5.66±.10	13.76±.19	18.81±.42	15.16±.24
	Max	14.15±.26	19.36±.24	15.59±.23	5.27±.38	6.99±.40	5.66±.37	13.90±.26	18.92±.25	15.29±.24
	HSSC+copy	14.70±.23	20.29±.33	16.27±.25	6.18±.14	8.38±.24	6.71±.17	14.46±.23	19.88±.30	15.98±.25
	Max+copy	14.62±.22	20.52±.59	16.29±.31	5.92±.16	8.19±.15	6.48±.11	14.37±.19	20.09±.59	15.99±.29
	Dual-view	14.83±.30	20.76±.41	16.50±.30	6.17±.31	8.57±.23	6.75±.26	14.57±.28	20.30±.36	16.19±.26
Home	PGNet	14.82±.19	20.53±.22	16.44±.19	6.28±.13	8.83±.15	6.90±.13	14.64±.18	20.23±.21	16.23±.18
	C.Transformer	13.75±.13	19.35±.15	15.36±.13	5.44±.13	7.70±.14	6.01±.13	13.58±.12	19.06±.15	15.17±.13
	HSSC	14.71±.17	20.29±.26	16.28±.20	5.85±.14	8.10±.19	6.39±.15	14.51±.17	19.97±.26	16.05±.20
	Max	14.85±.13	20.35±.14	16.39±.13	5.97±.08	8.26±.11	6.53±.09	14.65±.11	20.03±.12	16.15±.11
	HSSC+copy	14.93±.17	20.62±.17	16.54±.17	6.34±.11	8.87±.14	6.95±.11	14.75±.17	20.32±.17	16.33±.17
	Max+copy	14.92±.13	20.57±.17	16.52±.14	6.33±.11	8.84±.14	6.94±.11	14.72±.12	20.25±.17	16.30±.13
	Dual-view	15.18±.11	20.96±.09	16.81±.09	6.57±.11	9.19±.14	7.21±.12	15.00±.10	20.65±.07	16.60±.08

Table 3: Sentiment classification results on the four datasets.

Method	Sports		Movies		Toys		Home	
	M. F ₁	B. Acc	M. F ₁	B. Acc	M. F ₁	B. Acc	M. F ₁	B. Acc
HSSC	53.49±.83	57.03±.79	60.67±.40	63.21±.185	54.24±1.05	56.83±1.81	58.51±.46	60.78±.59
Max	53.27±1.70	56.48±1.11	60.66±.39	62.98±.33	55.02±.49	56.57±1.52	58.31±.78	60.67±.80
BiGRU+Attention	54.21±1.21	57.80±1.38	61.14±.38	63.26±.30	53.54±2.84	57.95±.51	59.32±.72	61.75±.70
DARLM	49.60±1.85	58.56±1.21	57.75±.43	62.58±.42	50.58±.45	55.09±1.07	54.49±1.82	59.30±.61
HSSC+copy	53.14±1.33	56.30±.45	60.68±.37	63.06±.20	54.38±.86	57.40±1.03	58.78±.80	60.77±.66
Max+copy	53.95±1.19	56.58±.62	60.60±.36	63.14±.26	53.52±1.60	58.22±.80	58.85±.41	60.70±.21
Dual-view	56.31±.67	59.30±.89	62.00±.19	64.46±.34	55.70±.86	58.30±.66	60.73±.55	62.87±.43

6.4 Case Studies (RQ4)

We conduct case studies to analyze the readability of the generated review summary and the sentiment consistency between the predicted sentiment labels and summaries. Table 7 compares the predicted sentiment labels and the generated summaries from the HSSC+copy model² and our dual-view model on the testing sets. We use ‘oracle’ to denote a model that always outputs the ground-truth. From row ‘a’ to row ‘d’ of Table 7, we observe that the sentiment labels and the summaries predicted by the HSSC+copy

²HSSC+copy is a strong baseline which enhances the state-of-the-art model HSSC [24] with a copy mechanism [34].

model are not consistent with each other. For example, in row ‘a’ of the table, the sentiment label predicted by HSSC+copy is 3, which indicates a neutral sentiment and matches the ground-truth. However, its generated summary (“not worth the money”) conveys a negative sentiment. In row ‘d’ of the table, the HSSC+copy model generates a summary with a positive sentiment that is similar to the ground-truth, but it predicts a sentiment label of 3, which is not consistent with the positive sentiment of the generated summary. On the other hand, the sentiment labels and the review summaries predicted by our dual-view model are sentimentally consistent to each other on these rows. Moreover, the HSSC+copy model sometimes

Table 4: Ablation study on the Sports dataset. “Full” indicates our full model. “-I” means the inconsistency loss is removed. “-A” means we replace the attention based classifier with a max pooling based classifier. “-R” means we remove the residual connection in the encoder part and only use one GRU layer as the encoder. “-C” indicates the copy mechanism is removed. RG-* is the F_1 -Measure of ROUGE-*

	RG-1	RG-2	RG-L	M. F_1	B. Acc
Full	16.79	6.98	16.55	56.31	59.30
-I	16.38	6.70	16.15	55.95	58.57
-A	16.50	6.81	16.25	55.36	57.14
-R	16.31	6.64	16.06	54.38	58.34
-C	15.33	5.72	15.10	55.92	58.50

Table 5: The macro F_1 scores of the source-view, summary-view, and merged sentiment classifiers on the four datasets.

Classifiers	Sports	Movies	Toys	Home
Source-view	56.31	62.00	55.70	60.73
Summary-view	55.56	61.55	55.02	60.48
Merged	56.28	61.90	55.55	60.92

Table 6: The macro F_1 scores of the source-view, summary-view, and merged sentiment classifiers with and without teacher-forcing on Sports validation and testing datasets. “w/ TF” means we apply teacher forcing when generating summaries. “w/o TF” indicates teacher forcing is removed.

Classifiers	Validation Set		Testing Set	
	w/ TF	w/o TF	w/TF	w/o TF
Source-view	54.81	54.81	56.31	56.31
Summary-view	56.86	54.26	58.62	55.56
Merged	56.06	54.86	57.44	56.28

omits sentiment words in the generated summaries, which makes the summaries less informative. For example, in row “f” of the table, the summary generated by HSSC+copy omits the sentiment word “paramount”. Thus, it does not reflect the opinion that the user likes the product very much, while the summary generated by our model contains “paramount” which expresses the consumer’s opinion accurately.

7 CONCLUSION

We propose a novel dual-view model with inconsistency loss to jointly improve the performance of review summarization and sentiment classification. Compared to previous work, our model has the following two advantages. First, it encourages the sentiment information in the decoder states to be similar to that in the review context representation, which assists the decoder to generate a summary that has the same sentiment tendency as the review. Second, the source-view and summary-view sentiment classifiers can learn from each other to improve the sentiment classification performance. Experiment results demonstrate that our model achieves

Table 7: Samples of the predicted sentiment labels and review summaries from different models. “Oracle” denotes a model that always outputs the ground-truth. The predicted sentiment labels (**boldfaced and italicized**) are placed before each generated summary. The sentiment labels ranges from 1 to 5, where 5 (1) denotes the most positive (negative) sentiment attitude.

Id	Model	Predicted Sentiment Label and Summary
a	Oracle	3 : you get what you pay for .
	HSSC+copy	3 : not worth the money .
	Dual-view	3 : you get what you pay for .
b	Oracle	1 : do not buy it .
	HSSC+copy	1 : this thing is okay .
	Dual-view	1 : not worth it .
c	Oracle	1 : worst toaster i have ever owned .
	HSSC+copy	2 : worst toaster i have ever owned .
	Dual-view	1 : worst toaster i have ever owned .
d	Oracle	5 : best for side/stomach sleepers .
	HSSC+copy	3 : good for side/stomach sleepers .
	Dual-view	5 : comfortable for side/stomach sleepers .
e	Oracle	5 : makes great ice cream !
	HSSC+copy	5 : ice cream maker .
	Dual-view	5 : great ice cream attachment !
f	Oracle	5 : paramount stockton 5-piece daybed ensemble , twin leggett & platt - home textiles .
	HSSC+copy	5 : 5-piece daybed ensemble .
	Dual-view	5 : paramount stockton 5-piece daybed ensemble .
g	Oracle	4 : works great for less money .
	HSSC+copy	5 : it ’s a rangefinder .
	Dual-view	4 : great for the price .

better performance than the state-of-the-art models for the task of joint review summarization and sentiment classification.

ACKNOWLEDGMENTS

The work described in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 2300174 (Collaborative Research Fund, No. C5026-18GF)). We would like to thank the four anonymous reviewers for their comments.

REFERENCES

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR, May 7-9, 2015*. <http://arxiv.org/abs/1409.0473>
- [2] Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. 2016. Neural Combinatorial Optimization with Reinforcement Learning. *CoRR* abs/1611.09940 (2016). arXiv:1611.09940 <http://arxiv.org/abs/1611.09940>
- [3] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. 2010. The Balanced Accuracy and Its Posterior Distribution. In *ICPR, 23-26 August 2010*. 3121–3124. <https://doi.org/10.1109/ICPR.2010.764>
- [4] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2017. Improving Multi-Document Summarization via Text Classification. In *AAAI, February 4-9, 2017*. 3053–3059. <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14525>
- [5] Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *ACL, July, 2018*. Association for Computational Linguistics, Melbourne, Australia, 675–686. <https://doi.org/10.18653/v1/P18-1063>

- [6] Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. 2017. Aspect-level Sentiment Classification with HEAT (HiERarchical ATtention) Network. In *CIKM, November, 2017*. ACM, 97–106. <https://doi.org/10.1145/3132847.3133037>
- [7] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP, Oct, 2014*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. <http://aclweb.org/anthology/D/D14/D14-1179.pdf>
- [8] Lea Frermann and Alexandre Klementiev. 2019. Inducing Document Structure for Aspect-based Summarization. In *ACL, July, 2019*. Association for Computational Linguistics, Florence, Italy, 6263–6273. <https://doi.org/10.18653/v1/P19-1630>
- [9] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions. In *COLING, August, 2010*. Coling 2010 Organizing Committee, Beijing, China, 340–348. <https://www.aclweb.org/anthology/C10-1039/>
- [10] Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, and Masaru Kitsuregawa. 2013. Modeling User Leniency and Product Popularity for Sentiment Classification. In *IJCNLP, October, 2013*. Asian Federation of Natural Language Processing, Nagoya, Japan, 1107–1111. <https://www.aclweb.org/anthology/I13-1156/>
- [11] Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-Up Abstractive Summarization. In *EMNLP, October 31 - November 4, 2018*. 4098–4109. <https://www.aclweb.org/anthology/D18-1443/>
- [12] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bitu Nejat. 2014. Abstractive Summarization of Product Reviews Using Discourse Structure. In *EMNLP, October, 2014*. Association for Computational Linguistics, Doha, Qatar, 1602–1613. <https://www.aclweb.org/anthology/D14-1168/>
- [13] Andrew Goldberg and Xiaojin Zhu. 2006. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *ACL, TextGraphs Workshop, June, 2006*. Association for Computational Linguistics, New York City, 45–52.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR, June 27-30, 2016*. 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [16] Vikrant Hole and Mukta Takalikar. 2013. Real time tweet summarization and sentiment analysis of game tournament. *International Journal of Science and Research* 4, 9 (2013), 1774–1780.
- [17] Wan Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss. In *ACL, July, 2018*. Association for Computational Linguistics, Melbourne, Australia, 132–141. <https://doi.org/10.18653/v1/P18-1013>
- [18] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD, August 22-25, 2004*. ACM, New York, NY, USA, 168–177. <https://doi.org/10.1145/1014052.1014073>
- [19] John D Kelleher, Brian Mac Namee, and Aoife D'arcy. 2015. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT Press.
- [20] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR, May 7-9, 2015*. <http://arxiv.org/abs/1412.6980>
- [21] Junjie Li, Haoran Li, and Chengqing Zong. 2019. Towards Personalized Review Summarization via User-Aware Sequence Network. In *AAAI, January 27 - February 1, 2019*. 6690–6697. <https://doi.org/10.1609/aaai.v33i01.33016690>
- [22] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. In *SIGIR, August 7-11, 2017*. 345–354. <https://doi.org/10.1145/3077136.3080822>
- [23] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* (2004).
- [24] 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 *IJCAI, July 13-19, 2018*. 4251–4257. <https://doi.org/10.24963/ijcai.2018/591>
- [25] Vinod L Mane, Suja S Panicker, and Vidya B Patil. 2015. Summarization and sentiment analysis from user health posts. In *ICPC, 2015*. IEEE, 1–4.
- [26] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL: System Demonstrations, June, 2014*. Association for Computational Linguistics, Baltimore, Maryland, 55–60. <https://www.aclweb.org/anthology/P14-5010/>
- [27] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR, 2015*. ACM, New York, NY, USA, 43–52. <https://doi.org/10.1145/2766462.2767755>
- [28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NEURIPS, December 5-8, 2013*. Curran Associates, Inc., 3111–3119. <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality>
- [29] Cataldo Musto, Gaetano Rossiello, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2019. Combining text summarization and aspect-based sentiment analysis of users' reviews to justify recommendations. In *RecSys, September, 2019*. ACM, 383–387. <https://doi.org/10.1145/3298689.3347024>
- [30] Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *ACL, June, 2005*. Association for Computational Linguistics, Ann Arbor, Michigan, 115–124. <https://www.aclweb.org/anthology/P05-1015/>
- [31] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *EMNLP, July, 2002*. Association for Computational Linguistics, Philadelphia, PA, USA, 79–86. <https://www.aclweb.org/anthology/W02-1011/>
- [32] Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. 2010. The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns. In *COLING, August, 2010*. Coling 2010 Organizing Committee, Beijing, China, 913–921. <https://www.aclweb.org/anthology/C10-1103/>
- [33] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence Level Training with Recurrent Neural Networks. In *ICLR, May 2-4, 2016*. <http://arxiv.org/abs/1511.06732>
- [34] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL, July, 2017*. Association for Computational Linguistics, Vancouver, Canada, 1073–1083. <https://doi.org/10.18653/v1/P17-1099>
- [35] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *EMNLP, September, 2015*. Association for Computational Linguistics, Lisbon, Portugal, 1422–1432. <https://doi.org/10.18653/v1/D15-1167>
- [36] Yufei Tian, Jianfei Yu, and Jing Jiang. 2019. Aspect and Opinion Aware Abstractive Review Summarization with Reinforced Hard Typed Decoder. In *CIKM, November, 2019*. ACM, 2061–2064. <https://doi.org/10.1145/3357384.3358142>
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NEURIPS, 4-9 December 2017*. 5998–6008. <http://papers.nips.cc/paper/7181-attention-is-all-you-need>
- [38] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order Matters: Sequence to sequence for sets. In *ICLR, May 2-4, 2016*. <http://arxiv.org/abs/1511.06391>
- [39] Lu Wang and Wang Ling. 2016. Neural Network-Based Abstract Generation for Opinions and Arguments. In *NAACL, June, 2016*. Association for Computational Linguistics, San Diego, California, 47–57. <https://www.aclweb.org/anthology/N16-1007/>
- [40] Wenting Xiong and Diane J. Litman. 2014. Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews. In *COLING, August, 2014*. Technical Papers. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 1985–1995. <https://www.aclweb.org/anthology/C14-1187/>
- [41] Chang Xu, Dacheng Tao, and Chao Xu. 2013. A Survey on Multi-view Learning. *CoRR* abs/1304.5634 (2013). [arXiv:1304.5634](http://arxiv.org/abs/1304.5634) <http://arxiv.org/abs/1304.5634>
- [42] Min Yang, Qiang Qu, Ying Shen, Qiao Liu, Wei Zhao, and Jia Zhu. 2018. Aspect and Sentiment Aware Abstractive Review Summarization. In *COLING, August, 2018*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 1110–1120.
- [43] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *NAACL, June, 2016*. Association for Computational Linguistics, San Diego, California, 1480–1489. <https://doi.org/10.18653/v1/N16-1174>
- [44] Yichun Yin, Yangqiu Song, and Ming Zhang. 2017. Document-Level Multi-Aspect Sentiment Classification as Machine Comprehension. In *EMNLP, September, 2017*. Association for Computational Linguistics, Copenhagen, Denmark, 2044–2054. <https://doi.org/10.18653/v1/D17-1217>
- [45] Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 8, 4 (2018). <https://doi.org/10.1002/widm.1253>
- [46] Qianrong Zhou, Xiaojie Wang, and Xuan Dong. 2018. Differentiated Attentive Representation Learning for Sentence Classification. In *IJCAI, July 13-19, 2018*. 4630–4636. <https://doi.org/10.24963/ijcai.2018/644>
- [47] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based LSTM Network for Cross-Lingual Sentiment Classification. In *EMNLP, November, 2016*. Association for Computational Linguistics, Austin, Texas, 247–256. <https://doi.org/10.18653/v1/D16-1024>