Aspect and Sentiment Aware Abstractive Review Summarization

Min Yang[1], Qiang Qu[1]*, Ying Shen[2], Qiao Liu[3], Wei Zhao[1], Jia Zhu[4]
[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2] Peking University Shenzhen Graduate School
[3] University of Electronic Science and Technology of China
[4] South China Normal University
{min.yang,qiang,wei.zhao}@siat.ac.cn, shenying@pkusz.edu.cn
qliu@uestc.edu.cn, jzhu@m.scnu.edu.cn

Abstract

Review text has been widely studied in traditional tasks such as sentiment analysis and aspect extraction. However, to date, no work is towards the end-to-end abstractive review summarization that is essential for business organizations and individual consumers to make informed decisions. This work takes the lead to study the aspect/sentiment-aware abstractive review summarization in an end-to-end manner without hand-crafted features and templates by exploring the encoder-decoder framework and multi-factor attentions. Specifically, we propose a mutual attention mechanism to interactively learns the representations of context words, sentiment words and aspect words within the reviews, acted as an encoder. The learned sentiment and aspect representations are incorporated into the decoder to generate aspect/sentiment-aware review summaries via an attention fusion network. In addition, the abstractive summarizer is jointly trained with the text categorization task, which helps learn a category-specific text encoder, locating salient aspect information and exploring the variations of style and wording of content with respect to different text categories. The experimental results on a real-life dataset demonstrate that our model achieves impressive results compared to other strong competitors.

1 Introduction

User generated reviews on products are expanding rapidly with the emergence and advancement of e-commerce. These reviews are valuable to business organizations for improving their products and to individual consumers for making informed decisions. Unfortunately, reading though all the product reviews is hard, especially for the reviews that are lengthy and have low readability. It is therefore essential to provide coherent and concise summaries of user generated reviews. In this paper, we focus on generating abstractive summaries of product reviews. Abstractive text summarization is the task of generating a short and concise summary that captures the salient ideas of the source text. The generated summaries potentially contain new phrases and sentences that may not appear in the source text. Inspired by recent success of sequence-to-sequence (seq2seq) model in statistical machine translation, most abstractive summarization systems employ seq2seq framework to generate summaries (Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017). In general, the seq2seq model firstly uses an encoder to convert the input text as a vector representation, and then it feeds this representation into a decoder to generate summary.

Despite the remarkable progress of previous studies, generating aspect/sentiment-aware summaries of product reviews remains a challenge in real-world for two reasons. (i) First, neural sequence-to-sequence models tend to generate trivial and generic summary, often involving high-frequency phrases. These summaries cannot capture the aspect and sentiment information from the product reviews which play a vital role in helping customers to make quick and informed decisions on certain products (Ly et al., 2011). (ii) According to what we observe, summary styles and words in different categories can significantly vary. However, existing methods apply a uniform model to generate text summaries for

* Corresponding author.

Proceedings of the 27th International Conference on Computational Linguistics, pages 1110–1120
Santa Fe, New Mexico, USA, August 20-26, 2018.

the source documents in different categories, which easily miss or under represent salient aspects of the documents.

To alleviate the aforementioned limitations, we design a Multi-factor attention fusion network for aspect/sentiment-aware Abstractive Review Summarization (MARS). Our model exploits the recent success of the encoder-decoder framework to generate aspect/sentiment-aware review summaries. Specifically, a mutual attention mechanism is proposed to capture the correlation of context words, sentiment words and aspect words, which interactively learns attentions in the three kinds of words and generates the representations for contexts, sentiments and aspects separately. The text encoder is regularized with the co-training to perform an additional task of text categorization. The purpose of co-training is not to achieve the best performance on the text categorization (auxiliary task), but rather to compensate for the missing regularization requirement of abstractive summarization in the standard framework, learning a category-specific text encoder and improving the the quality of locating salient aspect information of the review. In addition, we explore three kinds of attentions (i.e., semantic attention, sentiment attention and aspect attention) to selectively attend to the context information when decoding summaries. Finally, we employ a reinforcement learning technique to maximize long-term rewards and address the exposure bias issue (Ranzato et al., 2015).

We summarize our main contributions as follows:

- We leverage text categorization task to learn better category-specific review representation for summarization. The variation of style and wording of summaries with respect to different text categorization are explored.
- We exploit the coordination of context words, sentiment words and aspect words via the mutual attention mechanism to learn aspect/sentiment-aware review representation. With this design, our model can also well represent the collocative sentiment and aspect words, which are helpful to learn sentiment and aspect attentions during decoding.
- We explore three kinds of attentions (i.e., semantic attention, sentiment attention and aspect attention) to selectively attend to the context information when decoding summaries.
- We employ the reinforcement learning technique (i.e., policy gradient) to directly optimize the model with respect to the non-differentiable ROUGE scores, moderating the exposure bias issue.
- The experimental results show that our model outperforms the competitors from both quantitative and qualitative perspectives.

2 Related work

In general, existing text summarization approaches can be categorized as extractive and abstractive. The extractive summarization copies representative sentences from the input (Zhang et al., 2012), while the abstractive summarization generates new phrases, possibly rephrasing or using words that are not in the original text (Rush et al., 2015). In this paper, we focus on abstractive text summarization systems.

Inspired by the recent success of the encoder-decoder framework in statistical machine translation, there has been increasing interest in generalizing the neural language model to the field of abstractive summarization (Rush et al., 2015; Chopra et al., 2016; Chen et al., 2016; Nallapati et al., 2016; See et al., 2017). For example, Rush et al. (2015) were the first to apply attention-based encoder-decoder model to abstractive text summarization, achieving state-of-the-art performance two sentence-level summarization datasets. Nallapati et al. (2016) proposed off-the-shelf attention encoder-decoder RNN that captured hierarchical document structure and identified the key sentences and keywords in the document. See et al. (2017) proposed a hybrid pointer-generator network that allowed both copying words from the source text via pointing, and generating words from a fixed vocabulary.

Several recent studies attempted to integrate the encoder-decoder RNN and reinforcement learning paradigms for abstractive summarization, taking advantages of both (Paulus et al., 2017; Liu et al., 2018). For example, Paulus et al. (2017) combined the maximum-likelihood cross-entropy loss with

rewards from policy gradient reinforcement learning to reduce exposure bias. Liu et al. (2018) proposed an adversarial process for abstractive text summarization, in which the generator is built as an agent of reinforcement learning.

There are also some studies working on summarization of product reviews (Li et al., 2010; Gerani et al., 2014; Di Fabbrizio et al., 2014; Gerani et al., 2016; Yu et al., 2016; Mason et al., 2016). For example, Gerani et al. (2014) proposed an abstractive summarization system to product reviews by applying a template-based NLG framework and taking advantage of the discourse structure of reviews. Yu et al. (2016) proposed a phrase-based approach which leveraged phrase properties to choose a subset of optimal phrases for generating the final summary.

Different from the previous work, this study focuses on generating sentiment/aspect-aware abstractive review summarization that may better fit users' needs by using encoder decoder framework with sentiment/aspect attentions.

3 Methodology

This section defines the key notations and briefly formulates the problem of this study. We suppose that a review x consists of k words $x^c = [w_1^c, w_2^c, ..., w_k^c]$, n aspect words $x^t = [w_1^t, w_2^t, ..., w_n^t]$, and m sentiment words $x^s = [w_1^s, w_2^s, ..., w_m^s]$. To prevent conceptual confusion, we use superscripts "s", "t" and "c" to indicate the variables that are related to sentiment words, aspect words and content, respectively. Each review x in the corpus has a category label y and a corresponding reference summary $Z = [z_1, z_2, \ldots, z_T]$, where T is the length of the reference summary.

Our model MARS consists of two tasks: the abstractive review summarization task and the text categorization task, both working on a shared document encoding layer. In this section, we elaborate the main components of MARS in detail.

3.1 LSTM Encoder

This section introduces our Mutual Attention Network (MAN) to learn better sentiment, aspect and context representation via interactive learning. MAN utilizes the attention mechanism associated with the sentiment and aspect words to capture important information from the input review and learn the sentiment/aspect-aware review representation. Further, MAN makes use of the interactive information from the input review to supervise the modeling of the sentiment and aspect words which are helpful to capture important information in summary generation.

3.1.1 Initial Document Representation

Each word w in the review is mapped to a low-dimensional embedding $e \in \mathbb{R}^d$ through a word embedding layer, where d denotes the embedding dimensionality. Then, we employ three independent LSTM networks to obtain the hidden states of context words, the sentiment words, and the aspect words. Formally, given the input word embedding e_t at time step t, the hidden state h_t can be updated with the previous hidden state h_{t-1}, which is computed by

$$i_t = \sigma(W_i e_t + U_i h_{t-1} + b_i) \tag{1}$$
$$f_t = \sigma(W_f e_t + U_f h_{t-1} + b_f) \tag{2}$$
$$o_t = \sigma(W_o e_t + U_o h_{t-1} + b_o) \tag{3}$$
$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \tag{4}$$
$$\tilde{c} = \tanh(W_c e_t + U_c h_{t-1} + b_c) \tag{5}$$
$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where i_t, f_t, o_t, c_t are input gate, forget gate, output gate and memory cell, respectively. W and U denote weight matrices to be learned. b represents biases. σ is a sigmoid function and $\odot$ stands for element-wise multiplication. Hence, we can use the LSTM networks to obtain the hidden states $H^c = [h_1^c, h_2^c, ..., h_k^c] \in R^{k \times u}$ for context words, the hidden states $H^s = [h_1^s, h_2^s, ..., h_m^s] \in R^{m \times u}$ for sentiment words in the review, and the hidden states $H^t = [h_1^t, h_2^t, ..., h_n^t] \in R^{n \times u}$ for aspect words in the review, where u is

the size of hidden states for each LSTM unit. Then, we feed these hidden states to a mean-pooling layer to obtain the initial representation of context words, sentiment words, and aspect words in the review, respectively.

$$v^c = \sum_{i=1}^{k} h_i^s/k; \quad v^s = \sum_{i=1}^{m} h_i^c/m; \quad v^t = \sum_{i=1}^{n} h_i^t/n \tag{7}$$

3.1.2 Aspect/sentiment-aware Document Representation

Attention mechanism plays an important role in text modeling. Inspired by (Bahdanau et al., 2014; Ma et al., 2017), this section introduces the proposed mutual attention network (MAN) to learn a better sentiment-aware and aspect-specific document representation. In addition, the MAN model can also well represent the sentiment and aspect word representations. Formally, given the context word representations $[h_1^c, h_2^c, ..., h_k^c]$, the initial representations of sentiment and aspect words (i.e., v^s and v^t, respectively), the mutual attention mechanism generates the attention weight C_i of the context by

$$C_i = \frac{\exp(\rho([h_i^c; v^s; v^t]))}{\sum_{j=1}^{k} \exp(\rho([h_j^c; v^s; v^t]))} \tag{8}$$

where C_i indicates the importance of the i-th word in the context, and ρ is the attention function that calculates the importance of h_i^c in the context:

$$\rho([h_j^c; v^s; v^t]) = {U_c}^T \text{tanh}(W_c[h_j^c; v^s; v^t] + b_c) \tag{9}$$

where U_c and W_c are projection parameters to be learned, and b_c is the bias.

Only using the attention vector C cannot capture the interactive information of the context words and the aspect words (sentiment words), and lacks the ability of discriminating the importance of the words in the context. To make use of the interactive information between the context words and the aspect words (sentiment words), we also use the context words as attention source to attend to the aspect words (sentiment words). Similar to Eq. (8), we can calculate the attention vectors T and S for the aspect words and sentiment words as:

$$T_i = \frac{\exp(\rho([h_i^t; v^c; v^s]))}{\sum_{i=1}^{n} \exp(\rho([h_i^t; v^c; v^s]))} \tag{10}$$

$$S_i = \frac{\exp(\rho([h_i^s; v^c; v^t]))}{\sum_{i=1}^{n} \exp(\rho([h_i^s; v^c; v^t]))} \tag{11}$$

where ρ is the same as in Equation 9.

After computing the mutual attention vectors for the context words, aspect words and sentiment words, we can get the final context, aspect, sentiment representations emb^c, emb^s and emb^t based on the mutual attention vectors C, S and T by:

$$emb_x^c = \sum_{i=1}^{k} (C_i h_i^c), \quad emb_x^s = \sum_{i=1}^{m} (T_i h_i^s), \quad emb_x^t = \sum_{i=1}^{n} (S_i h_i^t) \tag{12}$$

Finally, we concatenate the context, aspect, sentiment representations to form the aspect/sentiment-aware review representation emb_x for review x:

$$emb_x = [emb_x^c, emb_x^t, emb_x^s] \tag{13}$$

3.2 Text Categorization

We feed the final document representation emb_x into a task-specific fully connected layer and a softmax classifier to predict the category distribution of the input document x:

$$\hat{y} = softmax(V_2 \cdot F_x), \; F_x = \text{tanh}(V_1 \cdot emb_x) \tag{14}$$

where V_1 and V_2 are projection parameters to be learned. We train this model by minimizing the cross-entropy between the predicted distribution $\hat{y}$ and the ground truth distribution y for each review in the training data:

$$J_{ML}^{\text{categ.}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{D}\mathbf{I}(y_i = j)\log(\hat{y}_i) \tag{15}$$

where θ is the set of parameters of our model, D is the number of categories, N is the number of reviews in the training set, $\mathbf{I}(\cdot)$ is an indicator such that $\mathbf{I}(\text{true}) = 1$ and $\mathbf{I}(\text{false}) = 0$.

3.3 Abstractive Review Summarization

The abstractive review summarization subtask shares the same review representation module (encoder) with the text categorization subtask. The generation of summary Z is performed by a LSTM decoder.

3.3.1 Category-specific Review Representation

To generate category-specific summaries, the review representation emb_x are transformed to category-specific review embedding which is expected to capture category characteristics. Inspired by (Dong et al., 2014; Cao et al., 2017), we develop a category-specific transformation process to make the transformed review embedding hold the category characteristics information. Formally, our model transforms the review embedding emb_x to a category-specific review embedding $cemb_x$ by

$$cemb_x = \tanh(\mathbf{W}_\mu \times emb_x) \tag{16}$$

where $\mathbf{W}_\mu \in \mathbb{R}^d$ is the transformation matrix, d is the dimensionality of the category-specific document embedding. Note that we define the same dimensionality for both the document embedding and the category-specific document embedding.

To make the transformed embedding capture category-specific information, we develop the category-specific transformation matrix $\mathbf{W}_\mu$ according to the predicted product category. We introduce $|C|$ sub-matrices $(\mathbf{W}_\mu^1, \cdots, \mathbf{W}_\mu^{|C|})$, with each directly corresponding to one product category. Based on the predicted category derived from Eq. 14, the category-specific transformation matrix $\mathbf{W}_\mu$ is computed as the weighted sum of these sub-matrices: $\mathbf{W}_\mu = \sum_{i=1}^{|C|} \hat{y}\mathbf{W}_\mu^i$. In this way, $\mathbf{W}_\mu$ is automatically biased to the sub-matrix of the predicted category.

3.3.2 LSTM Decoder

Inspired by (See et al., 2017), the pointer-generator network is adopted as the decoder to generate summaries. The pointer-generator network allows both copying words from input text via pointing (P_{vocab}), and generating words from a fixed vocabulary (P_{gen}). Thus, the pointer-generator has the ability to produce out-of-vocabulary (OOV) words.

The category-specific review representation $cemb_x$ is used to initialize the hidden states s_0 of LSTM decoder. On each step t of decoding, the decoder receives the word embedding of the previous word w_{t-1} (while training, this is the previous word of the reference summary; at test time it is the previous word emitted by the decoder) and update its hidden state s_t:

$$s_t = \text{LSTM}(s_{t-1}, c_t, w_{t-1}) \tag{17}$$

The attention mechanism is used to calculate the attention weights a_t and context vector c_t. Attention mechaism is expected to take both context-sentiment and context-aspect correlations into consideration. The enhanced context vector c_t is aggregated by the representation of those informative words (see Eq. 21). In this paper, we explore three kinds of attention: semantic attention, sentiment attention and aspect attention. Details of these three kinds of attention are described as follows.

Semantic Attention Semantic attention simply applies the context representation itself as attention source. Following (Shimaoka et al., 2017), we apply a multi-layer perceptron (MLP) to compute semantic attention weights as follows:

$$a_{t,i}^{semantic} = \tanh(W_{a_1} h_i^c + U_{a_1} s_t + b_{a_1}) \tag{18}$$

where W_{a_1} and U_{a_1} are parameter matrices, b_{a_1} is bias parameter. The attention computed for context words are independent of the aspect/sentiment words. Hence, it is difficult for semantic attention to focus on those context words that are highly related to the aspects and sentiments.

Sentiment Attention In order to capture the correlation between sentiment words and the context, we take sentiment word representation emb_x^s as attention source to compute sentiment attention weights:

$$a_{t,i}^{sentiment} = \tanh(emb_x^s W_{a_2} h_i^c + U_{a_2} s_t + b_{a_2}) \tag{19}$$

where W_{a_2} is a bi-linear parameter matrix, U_{a_2} is parameter matrix, b_{a_2} is bias parameter.

Aspect Attention Aspect attention applies aspect word representation emb_x^t as attention query, which is expected to capture the correlations between aspect words and context words.

$$a_{t,i}^{aspect} = \tanh(emb_x^t W_{a_3} h_i^c + U_{a_3} s_t + b_{a_3}) \tag{20}$$

where W_{a_3} is a bi-linear parameter matrix, U_{a_3} is parameter matrix, b_{a_3} is bias parameter.

Attention Fusion We define the attention fusion of the semantic attention, sentiment attention and aspect attention at timestep t as:

$$a_{t,i} = \text{softmax}(\lambda_1 a_{t,i}^{semantic} + \lambda_2 a_{t,i}^{sentiment} + \lambda_3 a_{t,i}^{aspect}), \quad c_t = \sum_{i=1}^{k} a_{t,i} h_i^c \tag{21}$$

where λ_1, λ_2 and λ_3 are hyper-parameters that determines the weights of the three kinds of attentions. We set λ_1 = 0.5, λ_2 = λ_3 = 0.25. Note that for the documents that do not contain sentiment words and aspect words, we use only the semantic attention to distinguish the important information.

The context vector c_t is then concatenated with the decoder state s_t and fed through a linear layer and a *softmax* layer to compute the output probability distribution over a vocabulary of words at the current state:

$$P_{vocab}(w_t) = \text{softmax}(V_{d_2}(V_{d_1}[s_t, c_t] + b_{d_1}) + b_{d_2}) \tag{22}$$

where V_{d_1}, V_{d_2}, b_{d_1} and b_{d_2} are learnable parameters. The number of rows in V_{d_2} represents the number of words in the vocabulary.

On top of the LSTM decoder, we adopt the copy mechanism (See et al., 2017) to integrate the attention attribution into the final vocabulary distribution which is defined as the interpolation between two probability distributions:

$$P(w_t) = p_{gen} P_{vocab}(w_t) + (1 - p_{gen}) \sum_{i:w_i = w_t} a_{t,i} \tag{23}$$

where $p_{gen} \in [0, 1]$ is the switch variable for controlling generating a word from the vocabulary or directly copying it from the original review. If w is an out-of-vocabulary (OOV) word, then $P_{vocab}(w)$ is zero; if w does not appear in the source review, then $\sum_{i:w_i=w} a_{t,i}$ is zero. p_{gen} can be defined as:

$$p_{gen} = \text{sigmoid}(U_{d_1}^T c_t + U_{d_2}^T s_t + U_{d_3}^T w_{t-1} + b_{gen}) \tag{24}$$

where vectors U_{d_1}, U_{d_2}, U_{d_3} and scalar b_{gen} are learnable parameters.

A common way of training a summary generation model is to estimate the parameters by minimizing the negative log-likelihood of the training data:

$$J_{\text{ML}}^{\text{sum}}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \log P(w_t) \tag{25}$$

3.4 Training of MARS Model

Overall, MARS consists of two subtasks, each has training objective. To make the document embedding sensitive to the category knowledge, we train these two related task simultaneously. The joint multi-task objective function is minimized by:

$$J_{\text{ML}}(\theta) = \gamma_1 J_{ML}^{\text{categ.}}(\theta) + \gamma_2 J_{\text{ML}}^{\text{sum}}(\theta) \tag{26}$$

where γ_1 and γ_2 are hyper-parameters that determines the weights of L_1 and L_2. Here, we set $\gamma_1 = 0.2$, $\gamma_2 = 0.8$.

3.4.1 Policy Gradient Reinforcement Learning for Summary Generation

However, the maximum likelihood estimation (MLE) method suffers from two main issues. First, the evaluation metric is different from the training loss. For example, in summarization generation systems, the encoder-decoder models are trained using the cross-entropy loss but they are typically evaluated at test time using discrete and non-differentiable metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). Second, the input of the decoder at each time step is often the previous ground-truth word during training. Nevertheless, when generating summaries in the testing phase, the input of the next time step is the previous word generated by the decoder. This exposure bias (Ranzato et al., 2015) leads to error accumulation at the testing phase. Once the model generates a "bad" word, the error will propagate and accumulate with the length of the sequence.

To alleviate the aforementioned issues when generating summaries, we also optimize directly for ROUGE-1 since it achieves best results among the alternatives such as METEOR (Lavie and Agarwal, 2007) and BLEU (Papineni et al., 2002), by using policy gradient algorithm, and minimize the negative expected rewards:

$$J_{\text{RL}}^{\text{sum}}(\theta) = (r(\hat{z}) - r(z^s)) \sum_{t}^{T} \log p(z_t^s | Z_{1:t-1}^s; X) \tag{27}$$

where $r(\hat{z})$ is the reward of a greedy decoding generated sequence $\hat{z}$, and $r(z^s)$ is the reward of sequence z^s generated by sampling among the vocabulary at each step.

After pre-training the proposed model by minimizing the joint ML objective (see Eq. 26), we switch the model to further minimize a mixed training objective, integrating the reinforcement learning objective $J_{\text{RL}}^{\text{sum}}(\theta)$ with the original multi-task loss $J_{\text{ml}}(\theta)$:

$$J_{\text{mixed}}(\theta) = \beta J_{\text{ML}}(\theta) + (1 - \beta) J_{\text{RL}}^{\text{sum}}(\theta) \tag{28}$$

where β is a hyper-parameter, and we set β=0.1.

4 Experimental Setup

4.1 Datasets Description

We evaluate our model on Amazon reviews dataset from Stanford Network Analysis Project (SNAP) (McAuley and Leskovec, 2013). The raw dataset consists of 34,686,770 Amazon reviews from 6,643,669 users spanning different kinds of products such as books, video games, food, music. Each review mainly contains product ID, user information, ratings, a plaintext review and a review summary. There are 82 tokens on average of reviews. Since the dataset is too large to process all at once on our local computer, we randomly choose 50,000 reviews from each of the books, food, electronics, movies, music and clothing categories. For each of the six product categories, we use 80% instances as the training data, 10% instances as the validation data, and the remaining are used for testing.

Sentiment Words Collection The sentiment lexicon used in this paper is the combination of three popular sentiment lexicons: HowNet (Dong and Dong, 2006), MPQA (Wilson et al., 2005) and Liu's Lexicon (Hu and Liu, 2004), which consists of 11,017 words in total.

Aspect Words Collection The aspect dictionary contains about 800 words provided by the experts of each product categories, including product names and the aspect words of the products. Following (Yang et al., 2014), we apply topic model to extend the aspect dictionary by using a minimal set of seed words as prior knowledge to discover a much richer lexicon.

4.2 Baseline Methods

In the experiments, we compare our model with several strong baseline methods:

SummaRuNNer A simple recurrent network based sequence classifier (Nallapati et al., 2017) which facilitates interpretable visualization of its decisions. It is an extractive summarization model.

ABS Attentional encoder decoder recurrent neural networks for abstractive text summarization proposed in (Nallapati et al., 2016).

PGC The pointer-generator coverage networks proposed in (See et al., 2017) which copies words from the source text via pointing, while retaining the ability to produce novel words through the generator.

DeepRL The deep reinforced model (ML+RL version) proposed in (Paulus et al., 2017), which introduce a new objective function by combining the maximum-likelihood cross-entropy loss with rewards from policy gradient reinforcement learning to reduce exposure bias.

GANsum The generative adversarial network for abstractive summarization (Liu et al., 2018).

4.3 Implementation Details

We use 100-dimensional word2vec (Mikolov et al., 2013) vectors pre-trained on English Wikipedia data to initialize the word embeddings, and all out-of-vocabulary words are initialized by sampling from the uniform distribution U(-0.25,0.25) We initialize the recurrent weight matrices of LSTMs as random orthogonal matrices, and all the bias vectors are initialized to zero. The hidden state size of each LSTM is 300. We conduct mini-batch (with size 64) training using Adadelta optimization algorithm. Other hyperparameters include: learning rate 0.01, L2 regularization 0.001, dropout 0.2. For the pointer-generator models, we use a vocabulary of 50k words for both training and text data.

5 Experimental Results

In this section, we compare our model with baseline methods from quantitative and qualitative perspectives.

5.1 Quantitative Evaluation

Following the same evaluation as in (Nallapati et al., 2016), we compare our model with baseline methods in terms of ROUGE-1, ROUGE-2, ROUGE-L and Human evaluation.

ROUGE-N is a widely used evaluation metric for summarization tasks. It measures the consistency between n-gram occurrences in the generated and reference summaries. ROUGE-L compares the longest common sequence between the reference summary and the generated summary. We summarize the ROUGE scores of our model and the baseline methods in Table 1. PGC consistently perform better than ABS. This may be because that the copy mechanism used in PGC can handle the out-of-vocabulary words. DeepRL and GANsum are better than PGC, because they utilize reinforcement learning to alleviate the exposure bias problem and optimize directly the evaluation metrics. **Our model** performs even better than the strong competitors by leveraging text categorization task and aspect/sentiment attentions to improve the summarization results. This verifies the effectiveness of our model for abstractive review summarization.

We also perform human evaluation to evaluate the readability and quality of the generated summaries. We randomly select 200 test examples from the dataset. For each review, three human evaluators are invited to rank each summary generated by all 6 models based on their readability, where 1 indicates the lowest level of readability while 6 indicates the highest level. The experimental results based on human annotations are summarized in Table 1 (fifth column). MARS achieves the best results. Specifically, MARS improves 14.9% on the human evaluation score over the best result of baseline methods (i.e., PGC) on the test data.

Method	ROUGE-1	ROUGE-2	ROUGE-L	Human Evaluation
SummaRuNNer	80.53	62.23	82.64	2.72
ABS	78.53	60.92	79.21	1.68
PGC	81.84	64.15	83.18	2.94
DeepRL	82.12	65.09	84.31	3.22
GANsum	82.64	66.12	84.31	3.42
MARS	**84.13**	**68.28**	**86.15**	**3.93**

Table 1: Quantitative evaluation results. All our ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script.

Method	ROUGE-1	ROUGE-2	ROUGE-L	Human Evaluation
MARS	**84.13**	**68.28**	**86.15**	**3.93**
w/o categorization	81.33	65.25	82.35	2.76
w/o sentiment attention	82.44	66.23	84.14	3.24
w/o aspect attention	83.05	66.09	84.85	3.35
w/o semantic attention	82.12	65.84	83.42	3.06

Table 2: Ablation test results.

5.2 Ablation Study

To investigate the effect of each component of the MARS model, we also perform the ablation test of MARS in terms of discarding text categorization, aspect attention, sentiment attention, and semantic attention. The results are reported in Table 2. For human evaluation, three human evaluators are invited to rank each summary generated by all 5 models based on their readability, where 1 indicates the lowest level of readability while 5 indicates the highest level. Generally, all three factors contribute, and text categorization contribute most. This is within our expectation since the the text categorization helps learn better category-specific review representations. In addition, it also helps the learning of aspect and sentiment representations. The aspect and sentiment attention also makes great contribution to abstractive review summarization, verifying that the aspect and sentiment information plays a vital role in review summaries.

Input summary: Our pup has experienced allergies in forms of hotspots and itching from other dog foods. The cheap 'you can buy it anywhere' food not only have crazy preservatives in them but can cause health problems for your pets. This food works wonders on reducing allergies and our dog loves the food. **Ground-truth summary:** Great allergy sensitive dog food, dogs love it. **Summary by GANsum:** Great foods loves. **Summary by MARS:** Great dog food for reducing allergies.
Input summary: The NOOKColor is awesome- it plays music, is expandable, displays gorgeous color, allows you to surf the internet (not optimally, but the ability is there), and is mostly easy to navigate. From me, it gets a solid 4 stars. The problem is that as an eReader, it's already overpriced, and the Amazon sellers who offer it are asking ridiculous amounts. It's cheaper (and ships free) from Barnes and Noble. Amazon is usually No1 at everything, including price, and I already own two Kindles (I would buy the Kindle in color if it came that way) but the price being charged for the NOOKColor on Amazon is absurd. **Ground-truth summary:** Awesome eReader, Ridiculously Priced from this seller. **Summary by GANsum:** NOOKColor is awesome music gorgeous color Amazon. **Summary by MCARS:** Awesome NOOKColor, absurd Amazon price.

Table 3: Examples summaries.

5.3 Qualitative Evaluation

To evaluate the proposed model qualitatively, we reported some generated summaries by different models. Due to the limitation of space, we randomly choose two generated summaries by GANsum and our model from test data for comparison. The results are reported in Table 3. We observe that MARS tends to generate more specific and meaningful summaries in response to the given reviews. For example, our model successfully catches the sentiment word "Awesome" for the product *eReader* and the sentiment

word "absurd" for the *price* aspect of the eReader. The advantage of our model comes from its capability of integrating category, sentiment and aspect information into the attention encoder-decoder model.

6 Conclusion

We proposed MARS to improve the performance of aspect/sentiment-aware abstractive review summarization. A mutual attention mechanism was employed to integrated the sentiment and aspect information into the encoder-decoder abstractive summarizer. In addition, MARS leveraged text categorization to improve the performance of summarization by learning a category-specific review representation. We also explored three kinds of attentions (i.e., semantic attention, sentiment attention and aspect attention) to selectively attend to the context information when decoding summaries. The experimental results on a real-life dataset showed that our model substantially outperformed the strong competitive methods.

Acknowledgement

The work was partially supported by the CAS Pioneer Hundred Talents Program, the National Science Foundation of China (No.61772117), the National Science Foundation of China (No.61750110516).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2017. Improving multi-document summarization via text classification. In *AAAI*, pages 3053–3059.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

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

Zhendong Dong and Qiang Dong. 2006. *HowNet and the Computation of Meaning*. World Scientific.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *The Annual Meeting of the Association for Computational Linguistics*, pages 49–54.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602–1613.

Shima Gerani, Giuseppe Carenini, and Raymond T Ng. 2016. Modeling content and structure for abstractive review summarization. *Computer Speech & Language*.

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

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd international conference on computational linguistics*, pages 653–661. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop on Text Summarization Branches Out*, volume 8.

Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. 2018. Generative adversarial network for abstractive text summarization. In *Association for the Advancement of Artificial Intelligence*.

Duy Khang Ly, Kazunari Sugiyama, Ziheng Lin, and Min-Yen Kan. 2011. Product review summarization from a deeper perspective. In *Annual international ACM/IEEE joint conference on Digital libraries*, pages 311–314. ACM.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4068–4074.

Rebecca Mason, Benjamin Gaska, Benjamin Van Durme, Pallavi Choudhury, Ted Hart, Bill Dolan, Kristina Toutanova, and Margaret Mitchell. 2016. Microsummarization of online reviews: An experimental study. In *AAAI*, pages 3015–3021.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *ACM conference on Recommender systems*, pages 165–172. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Association for the Advancement of Artificial Intelligence*, pages 3075–3081.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *The 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389. Association for Computational Linguistics.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083. Association for Computational Linguistics.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Min Yang, Dingju Zhu, M Rashed, and Kam-Pui Chow. 2014. Learning domain-specific sentiment lexicon with supervised sentiment-aware lda. *Frontiers in Artificial Intelligence and Applications*.

Naitong Yu, Minlie Huang, Yuanyuan Shi, et al. 2016. Product review summarization by exploiting phrase properties. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 1113–1124.

Lanbo Zhang, Yi Zhang, and Yunfei Chen. 2012. Summarizing highly structured documents for effective search interaction. In *SIGIR*. ACM.