

A Hierarchical End-to-End Model for Jointly Improving Text Summarization and Sentiment Classification

Shuming Ma¹, Xu Sun¹, Junyang Lin², Xuancheng Ren¹

¹MOE Key Lab of Computational Linguistics, School of EECS, Peking University

²School of Foreign Languages, Peking University

{shumingma, xusun, linjunyang, renxc}@pku.edu.cn

Abstract

Text summarization and sentiment classification both aim to capture the main ideas of the text but at different levels. Text summarization is to describe the text within a few sentences, while sentiment classification can be regarded as a special type of summarization which “summarizes” the text into a even more abstract fashion, i.e., a sentiment class. Based on this idea, we propose a hierarchical end-to-end model for joint learning of text summarization and sentiment classification, where the sentiment classification label is treated as the further “summarization” of the text summarization output. Hence, the sentiment classification layer is put upon the text summarization layer, and a hierarchical structure is derived. Experimental results on Amazon online reviews datasets show that our model achieves better performance than the strong baseline systems on both abstractive summarization and sentiment classification.¹

1 Introduction

Text summarization and sentiment classification are two important tasks in natural language processing. Text summarization aims at generating a summary with the major points of the original text. Compared with extractive summarization, which selects a subset of existing words in the original text to form the summary, abstractive summarization builds an internal semantic representation and then uses natural language generation techniques to create a summary that is closer to what a human might express. In this work, we mainly focus on the abstractive text summarization. Sentiment classification is to assign a sentiment label to determine the attitude or the opinion inside the text. It is also known as opinion mining, deriving the opinion or the attitude of a speaker. Both text summarization and sentiment classification aim at mining the main ideas of the text. Text summarization describes the text with words and sentences in a more specific way, while sentiment classification summarizes the text with labels in a more abstractive way.

Most of the existing models are built for either summarization or classification. For abstractive text summarization, the most popular model is the sequence-to-sequence model [Sutskever *et al.*, 2014; Rush *et al.*, 2015], where generating a short summary for the long source text can be regarded as a mapping between a long sequence and a short sequence. The model consists of an encoder and a decoder. The encoder encodes the original text into a latent representation, and the decoder generates the summary. Some recent abstractive summarization models are the variants of the sequence-to-sequence model [Chopra *et al.*, 2016; See *et al.*, 2017]. For sentiment classification, most of the recent work uses the neural network architecture [Kim, 2014; Tang *et al.*, 2015], such as LSTM or CNN, to generate a text embedding, and use a multi-layer perceptron (MLP) to predict the label from the embedding.

Some previous work [Hole and Takalikal, 2013; Mane *et al.*, 2015] proposes the models to produce both the summaries and the sentiment labels. However, these models train the summarization part and the sentiment classification part independently, and require rich, hand-craft features. There are also some work about the sentiment summarization [Titov and McDonald, 2008; Lerman *et al.*, 2009], which aim at extracting the sentences with a certain sentiment class from the original texts. These work only focuses on the summarization, and does not improve the sentiment classification.

In this work, we explore a first step towards improving both text summarization and sentiment classification within an end-to-end framework. We propose a hierarchical end-to-end model, which consists of a summarization layer and a sentiment classification layer. The summarization layer compresses the original text into short sentences, and the sentiment classification layer further “summarizes” the texts into a sentiment class. The hierarchical structure establishes a close bond between text summarization and sentiment classification, so that the two tasks can improve each other. After compressing the texts with summarization, it will be easier for the sentiment classifier to predict the sentiment labels of the shorter text. Besides, text summarization can point out the important and informative words, and remove the redundant and misleading information that is harmful to predict the sentiment. The sentiment classification can provide a more significant supervision signal for text summarization, and guides the summarization component to capture the sentiment ten-

¹The code is available at <https://github.com/lancopku/HSSC>

gency of the original text, which can improve the coherence between the short text and the original text.

We evaluate our proposed model on Amazon online reviews datasets. Experimental results show that our model achieves better performance than the strong baseline systems on both summarization and sentiment classification.

The contributions of this paper are listed as follows:

- We treat the sentiment classification as a special type of summarization, and perform sentiment classification and text summarization using a unified model.
- We propose a multi-view attention to obtain different representation of the texts for summarization and sentiment classification.
- Experimental results shows that our model outperforms the strong baselines that train the summarization and sentiment classification separately.

2 Proposed Model

In this section, we introduce our proposed model in details. In Section 2.1, we give the problem formulation. We explain the overview of our proposed model in Section 2.2. Then, we introduce each components of the model from Section 2.3 to Section 2.5. Finally, Section 2.6 gives the overall loss function and the training methods.

2.1 Problem Formulation

Given an online reviews dataset that consists of N data samples, the i -th data sample (x^i, y^i, l^i) contains an original text x^i , a summary y^i , and a sentiment label l^i . Both the original content x^i and the summary y^i are sequences of words:

$$x^i = \{x_1^i, x_2^i, \dots, x_{L_i}^i\}$$

$$y^i = \{y_1^i, y_2^i, \dots, y_{M_i}^i\}$$

where L_i and M_i denote the number of words in the sequences x^i and y^i , respectively. The label $l^i \in \{1, 2, \dots, K\}$ denotes the sentiment attitude of the original content x^i , from the lowest rating 1 to the highest rating K .

The model is applied to learn the mapping from the source text to the target summary and the sentiment label. For the purpose of simplicity, (x, y, l) is used to denote each data pair in the rest of this section, where x is the word sequence of an original text, y is the word sequence of the corresponding summary, and l is the corresponding sentiment label.

2.2 Model Overview

Figure 1 shows the architecture of our model. Our model consists of three components, which are the text encoder, the summary decoder, and the sentiment classifier. The text encoder compresses the original text x into the context memory h with a bi-directional LSTM. The summary decoder is a uni-directional LSTM, which then generates a summary vector $v^{(c)}$ and a sentiment vector $v^{(t)}$ sequentially with the attention mechanism by querying the context memory h . The summary vectors $v^{(c)}$ are used to generate the summary with a word generator. The sentiment vectors $v^{(t)}$ of all time steps

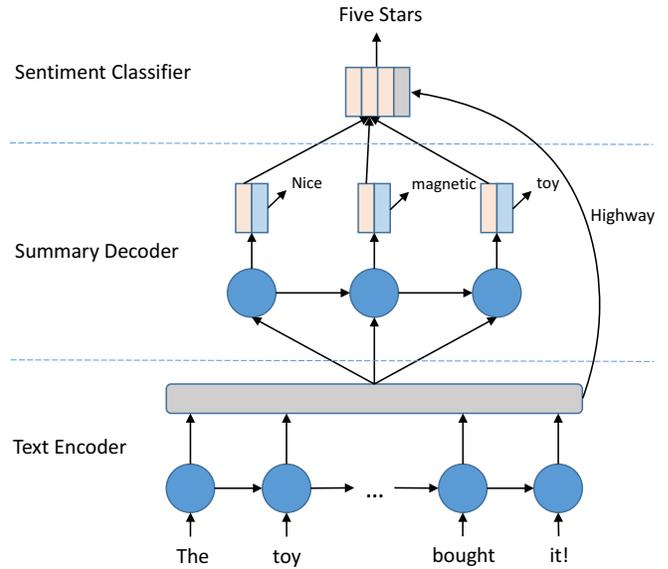


Figure 1: The overview of our model.

are collected and then fed into the sentiment classifier to predict the sentiment label. In order to capture the context information of the original text, we use the highway mechanism to feed the the context memory h as part of the input of the classifier. Therefore, the classifier predicts the label according to the sentiment vectors of the summary decoder and the context memory of the text encoder.

2.3 Text Encoder

The goal of the source text encoder is to provide a series of dense representation of the original text for the decoder and the classifier. In our model, the original text encoder is a bi-directional Long Short-term Memory Network (BiLSTM), which produces the context memory $h = \{h_1, h_2, \dots, h_L\}$ from the source text x :

$$\vec{h}_t = \vec{f}(x_t, \vec{h}_{t-1}) \tag{1}$$

$$\overleftarrow{h}_t = \overleftarrow{f}(x_t, \overleftarrow{h}_{t+1}) \tag{2}$$

$$h_t = \vec{h}_t + \overleftarrow{h}_t \tag{3}$$

where \vec{f} and \overleftarrow{f} are the forward and the backward functions of LSTM for one time step, \vec{h}_t and \overleftarrow{h}_t are the forward and the backward hidden outputs respectively, x_t is the input at the t -th time step, and L is the number of words in sequence x .

Although convolutional neural network (CNN) is also an alternative choice for the encoder, BiLSTM is more popular for the sequence-to-sequence learning of text generation tasks including abstractive text summarization. Besides, according to our experiments, BiLSTM achieves better performance in sentiment classification on our benchmark datasets. We give the details of the comparison of CNN and BiLSTM in Section 3.

2.4 Summary Decoder with Multi-View Attention

The goal of the summary decoder is to generate a series of summary words, and provides the summary information for

the sentiment classifier. In our model, the summary decoder consists of a uni-directional LSTM, a multi-view attention mechanism, and a word generator. The LSTM first generates the hidden output s_t conditioned on the historical information of the generated summary:

$$s_t = f(y_{t-1}, s_{t-1}) \quad (4)$$

where f is the function of LSTM for one time step, and y_{t-1} is the last generated words at t -th time step.

Given the hidden output s_t , we implement a multi-view attention mechanism to retrieval the summary information and the sentiment information from the context memory \mathbf{h} of the original text. The motivation of the multi-view attention is that the model should focus on different part of the original text for summarization and classification. For summarization, the attention mechanism should focus on the informative words that describe the main points best. For sentiment classification, the attention mechanism should focus on the words that contains the most sentimental tendency, such as “great”, “bad”, and so on. In implementation, the multi-view attention generates a summary vector $v^{(c)}$ for summarization:

$$v_t^{(c)} = \sum_{i=1}^N \alpha_{ti} h_i \quad (5)$$

$$\alpha_{ti} = \frac{e^{g(s_t, h_i)}}{\sum_{j=1}^N e^{g(s_t, h_j)}} \quad (6)$$

$$g(s_t, h_i) = \tanh(s_t^T W_t h_i) \quad (7)$$

where W_t is a trainable parameter matrix. Similar to the summary vector, the sentiment vector $v^{(t)}$ is also generated with the attention mechanism following Equation 5, 6, and 7, but has different trainable parameters. The multi-view attention can be regarded as two independent global attentions to learn to focus more on the summary aspect or the sentiment aspect.

Given the summary vector $v_t^{(c)}$, the word generator is used to compute the probability distribution of the output words at t -th time step:

$$p(y_t|x) = \text{softmax}(W_g v_t^{(c)} + b_g) \quad (8)$$

where W_g and b_g are parameters of the generator. The word with the highest probability is emitted as t -th word of the generated summary.

2.5 Summary-Aware Sentiment Classifier

After decoding the words until the end of the summary, the model collects the sentiment vectors of all time step:

$$\mathbf{v}^{(t)} = [v_1^{(t)}, v_2^{(t)}, \dots, v_M^{(t)}] \quad (9)$$

Then, we concatenate the summary sentiment vectors $\mathbf{v}^{(t)}$ and the original text representation \mathbf{h} , and perform a max-pooling operation to obtain a sentiment context vector \mathbf{r} , which we denote as a highway operation in Figure 1:

$$\begin{aligned} \mathbf{r} &= \max(\mathbf{v}^{(t)} \oplus \mathbf{h}) \\ &= \max([v_1^{(t)}, v_2^{(t)}, \dots, v_M^{(t)}, h_1, h_2, \dots, h_L]) \end{aligned} \quad (10)$$

where \oplus denotes the operation of concatenation along the first dimension, M is the number of words in the summary,

and L is the number of words in the original text. The sentiment context vector is then fed into the classifier to compute the probability distribution of the sentiment label $p(l|x)$. The classifier is a two-layer feed-forward network with RELU as the activation function. The label with the highest probability is the predicted sentiment label.

2.6 Overall Loss Function and Training

The loss function consists of two parts, which are the cross entropy loss of summarization and that of sentiment classification:

$$L_s = - \sum_t y_t \log p(y_t|x) \quad (11)$$

$$L_c = -l \log p(l|x) \quad (12)$$

where y_t and l are the ground truth of words and labels, and $p(y_t|x)$ and $p(l|x)$ are the probability distribution of words and labels, computed by Equation 8. We jointly minimize the two losses with Adam [Kingma and Ba, 2014] optimizer:

$$L = L_s + \lambda L_c \quad (13)$$

where λ is a hyper-parameter to balance two losses. We set $\lambda = 0.5$ in this work.

3 Experiments

In this section, we evaluate our model on the Amazon online review dataset, which contains the online reviews, summaries, and sentiment labels. We first introduce the datasets, evaluation metrics, and experimental details. Then, we compare our model with several popular baseline systems. Finally, we provide the analysis and the discussion of our model.

3.1 Datasets

Amazon SNAP Review Dataset (SNAP): This dataset is part of Stanford Network Analysis Project (SNAP)², and is provided by He and McAuley [2016]. The dataset consists of reviews from Amazon, and contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. It includes review content, product, user information, ratings, and summaries. We pair each review content with the corresponding summary and sentiment label. We select three domains of product reviews to construct three benchmark datasets, which are **Toys & Games**, **Sports & Outdoors**, and **Movie & TV**. We select the first 1,000 samples of each dataset as the validation set, the following 1,000 samples as the test set, and the rest as the training set.

3.2 Evaluation Metric

For abstractive summarization, our evaluation metric is ROUGE score [Lin and Hovy, 2003], which is popular for summarization evaluation. The metrics compare an automatically produced summary with the reference summaries, by computing overlapping lexical units, including unigram, bigram, trigram, and longest common subsequence (LCS). Following previous work [Rush *et al.*, 2015; Hu *et al.*, 2015], we

²<http://snap.stanford.edu/data/web-Amazon.html>

use ROUGE-1 (unigram), ROUGE-2 (bi-gram) and ROUGE-L (LCS) as the evaluation metrics in the reported experimental results.

For sentiment classification, the evaluation metric is per-label accuracy. We evaluate the accuracy of both five-class sentiment, of which the sentiment is classified into 5 class, and two-class sentiment, of which the sentiment is either positive or negative.

3.3 Experimental Details

Model Parameters

The vocabularies are extracted from the training sets, and the source contents and the summaries share the same vocabularies. We tune the hyper-parameters based on the performance on the validation sets.

We limit the vocabulary to 50,000 most frequent words appearing in the training set. We set the word embedding and the hidden size to 256, 512, and 512 for Toys, Sports, and Movies datasets, respectively. The word embedding is random initialized and learned from scratch. The encoder is a single-layer bidirectional LSTM, the decoder is a single-layer unidirectional LSTM, and the classifier is a two layer feed-forward network with a 512 hidden dimension. The batch size is 64, and we use dropout with probability $p = 0.2, 0.05, 0.0$ for Toys, Sports, and Movies datasets, respectively.

Model Training

We use the Adam [Kingma and Ba, 2014] optimization method to train the model. For the hyper-parameters of Adam optimizer, we set the learning rate $\alpha = 0.0003$, two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ respectively, and $\epsilon = 1 \times 10^{-8}$. Following Sutskever et al. [2014], we train the model for a total of 10 epochs, and start to halve the learning rate every half epoch after 5 epochs. We clip the gradients [Pascanu et al., 2013] to the maximum norm of 10.0.

3.4 Baselines

For abstractive summarization, our baseline model is the sequence-to-sequence model for abstractive summarization, following the previous work [Hu et al., 2015]. We denote the sequence-to-sequence model without the attention mechanism as **S2S**, and that with the attention mechanism as **S2S-att**.

For text classification, we compare our model with two baseline models: **BiLSTM** and **CNN**. For the two baseline models, the BiLSTM model uses a bidirectional LSTM with the dimension of 256 in each direction, and uses max pooling across all LSTM hidden states to get the sentence embedding vector, and then uses an MLP output layer with 512 hidden states to output the classification result. The CNN model uses the same scheme, but substitutes BiLSTM with 1 layer of convolutional network. During training we use 0.2 dropout on the MLP. We use Adam as the optimizer, with a learning rate of 0.001, and a batch size of 64. For BiLSTM, we also clip the norm of gradients to be 5.0. We searched hyperparameters in a wide range and find the aforementioned set of hyperparameters yield the highest accuracy.

The above baseline models only exploit part of the annotated data (either summaries or sentiment labels). For fairer

Toys & Games	RG-1	RG-2	RG-L
S2S [Hu et al., 2015]	14.05	2.47	15.75
S2S-att [Hu et al., 2015]	16.23	4.27	16.01
S2S-att + BiLSTM	16.32	4.43	16.27
HSSC (this work)	18.44	5.00	17.69
Sports & Outdoors	RG-1	RG-2	RG-L
S2S [Hu et al., 2015]	13.38	2.59	13.18
S2S-att [Hu et al., 2015]	15.70	3.61	15.53
S2S-att + BiLSTM	15.75	3.64	15.68
HSSC (this work)	17.85	4.77	17.59
Movie & TV	RG-1	RG-2	RG-L
S2S [Hu et al., 2015]	10.98	2.34	10.77
S2S-att [Hu et al., 2015]	12.17	3.08	11.77
S2S-att + BiLSTM	12.33	3.22	11.92
HSSC (this work)	14.52	4.84	13.42

Table 1: Comparison between our model and the sequence-to-sequence baseline for abstractive summarization on the Amazon SNAP test sets. The test sets include three domains: Toys & Games, Sports & Outdoors, and Movie & TV. RG-1, RG-2, and RG-L denote ROUGE-1, ROUGE-2, and ROUGE-L, respectively.

comparison, we also implement a joint model of S2S-att and BiLSTM (**S2S-att + BiLSTM**), and both the annotated labels of summaries and sentiments are used to train this baseline model. We compare our model with this model, in order to analyze the improvements of our model given exactly the same annotated data. In this baseline model, S2S-att and BiLSTM share the same encoder, and the S2S-att produces the summary with a LSTM decoder, while the BiLSTM predicts the sentiment label with a MLP. We tune the hyper-parameter on the validation set. We set the word embedding and the hidden size to 256, 512, and 512. The batch size is 64, and the dropout rate is $p = 0.15, 0.05, 0.1$ for Toys, Sports, and Movies datasets, respectively.

3.5 Results

We denote our Hierarchical Summarization and Sentiment Classification model as **HSSC**.

Abstractive Summarization

First, we compare our model with the sequence-to-sequence baseline on the Amazon SNAP test sets. We report the ROUGE F1 score of our model and the baseline models on the test sets. As shown in Table 1, our HSSC model has a large margin over both S2S and S2S-att models on all of the three test sets, which shows that the supervision of the sentiment labels improves the representation of the original text. Moreover, given exactly the same annotated data (summary + sentiment label), our HSSC model still has an improvement over the S2S-att + BiLSTM baseline, which indicates that HSSC learns a better representation for summarization. Overall, HSSC achieves the best performance in terms of ROUGE-1, ROUGE-2, and ROUGE-L over the three baseline models on the three test sets.

The summarization task on the online review texts is much more difficult and complicate, so the ROUGE scores on the SNAP dataset are lower than other summarization datasets,

Toys & Games	5-class	2-class
CNN	70.5	90.2
BiLSTM	70.7	90.9
BiLSTM + S2S-att	70.9	90.9
HSSC (this work)	71.9	91.8

Sports & Outdoors	5-class	2-class
CNN	72.0	91.5
BiLSTM	71.9	91.6
BiLSTM + S2S-att	72.1	91.9
HSSC (this work)	73.2	92.1

Movie & TV	5-class	2-class
CNN	66.9	86.0
BiLSTM	67.8	86.2
BiLSTM + S2S-att	68.0	86.6
HSSC (this work)	68.9	88.4

Table 2: Comparison between our model and the sequence-to-sequence baselines for sentiment classification on the Amazon SNAP test sets. The test sets include three domains: Toys & Games, Sports & Outdoors, and Movie & TV. 5-class and 2-class denote the accuracy of five-class sentiment and two-class sentiment, respectively.

such as DUC. The documents in DUC datasets are originally from news website, so the texts are formal, and the summaries in DUC are manually selected and well written. The SNAP dataset is constructed with the reviews on the amazon, and both the original reviews and the corresponding summaries are informal and full of noise.

Sentiment Classification

We compare our model with two popular sentiment classification methods, which are CNN and BiLSTM, on the Amazon SNAP test sets. We report the accuracy of five-grained sentiment and two-class sentiment on the test sets. As shown in Table 2, BiLSTM has a slightly improvement over the CNN baseline, showing that BiLSTM has a better performance to represent the texts on these datasets. Therefore, we select BiLSTM as the encoder of our model. HSSC obtains a better performance over the two widely-used baseline models on all of the test sets, mainly because of the benefit of more labeled data and better representation. What’s more, HSSC outperforms the S2S-att + BiLSTM baseline, showing that the information from summary decoder helps to predict the sentiment labels. Overall, HSSC achieves the best performance in terms of 5-class accuracy and 2-class accuracy over the three baseline models on the three test sets.

We have conducted significance tests based on t-test. The significance tests suggest that HSSC has a very significant improvement over all of the baselines, with $p \leq 0.001$ in all of ROUGE metrics for summarization in three benchmark datasets, $p \leq 0.005$ for sentiment classification in both Toys & Games and Movies & TV datasets, and $p \leq 0.01$ for sentiment classification in the Sports & Outdoors datasets.

3.6 Ablation Study

In order to analyze the effect of each components, We remove the components of multi-view and highway in order, and eval-

Toys & Games	5-class	RG-L
w/o Multi-View	70.9	16.47
w/o Highway	70.1	16.06
HSSC (Full Model)	71.9	17.69

Sports & Outdoors	5-class	RG-L
w/o Multi-View	72.0	16.36
w/o Highway	71.5	15.73
HSSC (Full Model)	73.2	17.59

Movie & TV	5-class	RG-L
w/o Multi-View	68.1	12.34
w/o Highway	67.7	12.01
HSSC (Full Model)	68.9	13.42

Table 3: Ablation study. 5-class denotes the accuracy of five-grained sentiment, and RG-L denotes ROUGE-L for summarization.

uate the performance of the rest model. We first remove the multi-view attention. As shown in Table 3, the model without multi-view attention has a drop of performance on both 5-class accuracy and ROUGE-L. It can be concluded that the multi-view attention improves the performance of both abstractive summarization and sentiment classification. We further remove the highway part, and find the highway component benefits not only the sentiment classification, but also the abstractive summarization. The benefit mainly comes from the fact that the gradient of the sentiment classifier can be directly propagated to the encoder, so that it learns a better representation of the original text for both classification and summarization.

3.7 Visualization of Multi-View Attention

As shown in Table 4, we present the heatmap of the attention scores of three examples. The multi-view attention allows the model to represent the text from the sentiment view and from the summary view. In order to analyze whether the multi-view attention captures the sentiment information and the summary information, we give the heatmap of the sentiment-view attention and the summary-view attention, respectively. We take the average of the attention scores in the decoder outputs at all time steps, and mark the high scores with deep color and the low scores with light color. From the table, we conclude that the sentiment-view attention focuses more on the sentimental words, e.g. “best”, “powerful”, “great”, “fun”, and “comfortable”. The summary-view attention concentrates on the informative words that best describes the opinion of the authors, e.g. “i think that this is one of the best movie”, and “a great book, very fun”. Moreover, the sentiment-view attention focuses more on the individual words, while the summary-view pays more attention on the word sequences. Besides, the sentiment-view attention and the summary-view attention share the focus on the informative words, showing the benefit from the multi-view attention.

4 Related Work

Rush et al. [2015] first proposes an abstractive based summarization model, which uses an attentive CNN encoder to

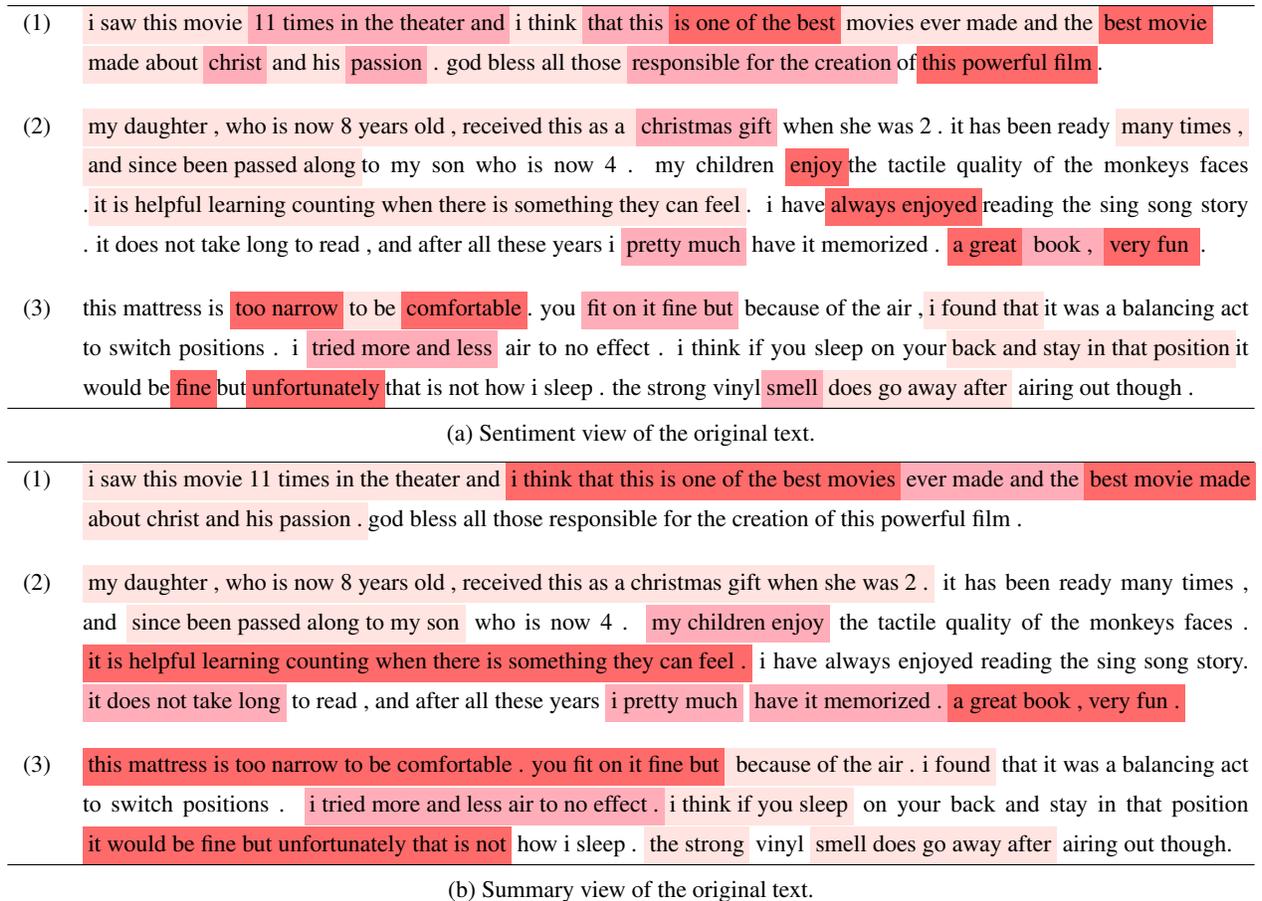


Table 4: Visualization of multi-view attention. Above is the heatmap of the sentiment-view attention, and below is the heatmap of the summary-view attention. Deeper colors means higher attention scores.

compress texts and a neural network language model to generate summaries. Chopra et al. [2016] explores a recurrent structure for abstractive summarization. To deal with out-of-vocabulary problem, Nallapati et al. [2016] proposes a generator-pointer model so that the decoder is able to generate words in source texts. Gu et al. [2016] also solves this issue by incorporating copying mechanism, allowing parts of the summaries to be copied from the source contents. See et al. [2017] further discusses this problem, and incorporates the pointer-generator model with the coverage mechanism. Hu et al. [2015] builds a large corpus of Chinese social media short text summarization. Chen et al. [2016] introduces a distraction based neural model, which forces the attention mechanism to focus on the difference parts of the source inputs. Ma et al. [2017] proposes a neural model to improve the semantic relevance between the source contents and the predicted summaries.

There are some work concerning with both summarization and sentiment classification. Hole and Takalikar [2013] and Mana et al. [2015] propose the models to produce both the summaries and the sentiment labels. However, these models train the summarization part and the sentiment classification part independently, and require rich hand-craft features. Some work has improved the summarization with the help

of classification. Cao et al. [2017] proposes a model to train the summary generator and the text classifier jointly, but only improves the performance of the text summarization. Titov and McDonald [2008] proposes a sentiment summarization method to extract the summary from the texts given the sentiment class. Lerman et al. [2009] builds a new summarizer by training a ranking SVM model over the set of human preference judgments, and improves the performance of sentiment summarization. Different from all of these works, our model improves both text summarization and sentiment classification, and does not require any hand-craft features.

5 Conclusions

In this work, we propose a model to generate both the sentiment labels and the human-like summaries, hoping to summarize the opinions from the coarse-grained sentiment labels to the fine-grained word sequences. We evaluate our proposed model on several online reviews datasets. Experimental results show that our model achieves better performance than the baseline systems on both abstractive summarization and sentiment classification.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. 61673028), National High Technology Research and Development Program of China (863 Program, No. 2015AA015404), and the National Thousand Young Talents Program. Xu Sun is the corresponding author of this paper.

References

- [Cao *et al.*, 2017] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Improving multi-document summarization via text classification. In *AAAI 2017*, pages 3053–3059, 2017.
- [Chen *et al.*, 2016] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. Distraction-based neural networks for modeling documents. In *IJCAI 2016*, New York, NY, July 2016. AAAI.
- [Cheng and Lapata, 2016] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *ACL 2016*, 2016.
- [Chopra *et al.*, 2016] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL HLT 2016*, pages 93–98, 2016.
- [Gu *et al.*, 2016] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL 2016*, 2016.
- [He and McAuley, 2016] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW 2016*, pages 507–517, 2016.
- [Hole and Takalikar, 2013] Vikrant Hole and Mukta Takalikar. Real time tweet summarization and sentiment analysis of game tournament. *International Journal of Science and Research*, 4(9):1774–1780, 2013.
- [Hu *et al.*, 2015] Baotian Hu, Qingcai Chen, and Fangze Zhu. LCSTS: A large scale chinese short text summarization dataset. In *EMNLP 2015*, pages 1967–1972, 2015.
- [Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP 2014*, pages 1746–1751, 2014.
- [Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [Lerman *et al.*, 2009] Kevin Lerman, Sasha Blair-Goldensohn, and Ryan T. McDonald. Sentiment summarization: Evaluating and learning user preferences. In *EACL 2009*, pages 514–522, 2009.
- [Lin and Hovy, 2003] Chin-Yew Lin and Eduard H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *HLT-NAACL 2003*, 2003.
- [Ma *et al.*, 2017] Shuming Ma, Xu Sun, Jingjing Xu, Houfeng Wang, Wenjie Li, and Qi Su. Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. In *ACL 2017*, pages 635–640, 2017.
- [Ma *et al.*, 2018] Shuming Ma, Xu Sun, Wei Li, Sujian Li, Wenjie Li, and Xuancheng Ren. Query and output: Generating words by querying distributed word representations for paraphrase generation. In *NAACL 2018*, 2018.
- [Mane *et al.*, 2015] Vinod L Mane, Suja S Panicker, and Vidya B Patil. Summarization and sentiment analysis from user health posts. In *Pervasive Computing (ICPC), 2015 International Conference on*, pages 1–4. IEEE, 2015.
- [Nallapati *et al.*, 2016] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos Santos, Çağlar Gülçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL 2016*, pages 280–290, 2016.
- [Pascanu *et al.*, 2013] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML 2013*, pages 1310–1318, 2013.
- [Rush *et al.*, 2015] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *EMNLP 2015*, pages 379–389, 2015.
- [See *et al.*, 2017] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *ACL 2017*, pages 1073–1083, 2017.
- [Sun *et al.*, 2017a] Xu Sun, Xuancheng Ren, Shuming Ma, and Houfeng Wang. meprop: Sparsified back propagation for accelerated deep learning with reduced overfitting. In *ICML 2017*, pages 3299–3308, 2017.
- [Sun *et al.*, 2017b] Xu Sun, Bingzhen Wei, Xuancheng Ren, and Shuming Ma. Label embedding network: Learning label representation for soft training of deep networks. *CoRR*, abs/1710.10393, 2017.
- [Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS 2014*, pages 3104–3112, 2014.
- [Takase *et al.*, 2016] Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. Neural headline generation on abstract meaning representation. In *EMNLP 2016*, pages 1054–1059, 2016.
- [Tang *et al.*, 2015] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP 2015*, pages 1422–1432, 2015.
- [Titov and McDonald, 2008] Ivan Titov and Ryan T. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL 2008*, pages 308–316, 2008.
- [Xu *et al.*, 2018a] Jingjing Xu, Xu Sun, Xuancheng Ren, Junyang Lin, Binzhen Wei, and Wei Li. Dp-gan: Diversity-promoting generative adversarial network for generating informative and diversified text. *CoRR*, abs/1802.01345, 2018.
- [Xu *et al.*, 2018b] Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL 2018*, 2018.