Attribute-aware Sequence Network for Review Summarization

Junjie Li[1], **Xuepeng Wang**[1], **Dawei Yin**[1], **Chengqing Zong**[2,3,4]
[1] Data Science Lab, JD.com, Beijing, China
[2] National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China
[4] CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing, China
{lijunjie46,wangxuepeng1}@jd.com,yindawei@acm.org,cqzong@nlpr.ia.ac.cn

Abstract

Review summarization aims to generate a condensed summary for a review or multiple reviews. Existing review summarization systems mainly generate summary only based on review content and neglect the authors' attributes (e.g., gender, age, and occupation). In fact, when summarizing a review, users with different attributes usually pay attention to specific aspects and have their own word-using habits or writing styles. Therefore, we propose an Attribute-aware Sequence Network (ASN) to take the aforementioned users' characteristics into account, which includes three modules: an attribute encoder encodes the attribute preferences over the words; an attribute-aware review encoder adopts an attribute-based selective mechanism to select the important information of a review; and an attribute-aware summary decoder incorporates attribute embedding and attribute-specific word-using habits into word prediction. To validate our model, we collect a new dataset *TripAtt*, comprising 495,440 attribute-review-summary triplets with three kinds of attribute information: gender, age, and travel status. Extensive experiments show that ASN achieves state-of-the-art performance on review summarization in both auto-metric ROUGE and human evaluation.

1 Introduction

Review summarization aims to generate a condensed summary for a review or multiple reviews[1]. Dominating studies can be divided into two groups: extractive and abstractive approaches. Extractive approaches (Hu and Liu, 2004; Ganesan, 2010) extract sentences or phrases from a review, while abstractive methods (Gerani et al., 2014; Wang and Ling, 2016; Yang et al., 2018a; Li et al., 2019; Gao et al., 2019) summarize a review by employing graph-based or sequence-to-sequence (S2S) models which can generate new phrases and sentences that do not appear in the review.

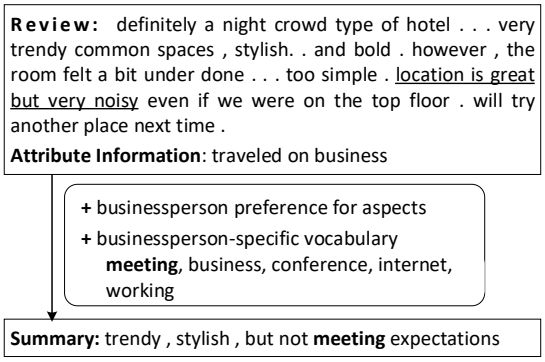

Figure 1: A review-summary pair posted by a businessperson in our dataset shows the effect of attribute information on summarizing review. Underlined words in the review indicate the important sentences that the businessperson care about when summarizing the review. Bold **word** in businessperson-specific vocabulary shows the businessperson's word-using habits may help to generate the summary.

Despite the remarkable progress of previous studies, they typically only focus on review content and neglect the attribute information of users who post these reviews (e.g., gender, age, and occupation). Actually, such information is vital for generating summaries, which contains the following characteristics. (1) People with different attributes may care about different aspects[2]. For example, when choosing hotels, business people may care about ***location*** and ***room*** more than ***price***, while solo travelers may prefer ***price*** more. Figure 1 presents a review posted by a businessper-

[1] Here, we focus on single-review summarization, and we leave adapting our model to multi-review summarization scenario to future work.

[2] Aspects refer to properties (attributes) of products or services, such as ***location*** and ***room*** for the hotel domain.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3000–3010, Hong Kong, China, November 3–7, 2019.

son. Although the hotel is trendy, stylish, and in a good location, it is noisy. The businessperson, therefore, summarizes that the hotel is not suitable for meeting. (2) People with different attributes have different word-using habits or writing styles to summarize a review. According to our statistics (Section 2.2), business people often summarize a review with words like "meeting", "business" and "conference", while solo travelers often utilize "budget", "inexpensive" to summarize their reviews. These attribute-specific words may help generate summaries. Figure 1 shows an example: without considering the attribute information "businessperson", it is hard to generate the word "meeting" when summarizing the review due to its missing. Intuitively, "meeting" belongs to businessperson-specific vocabulary, and such an attribute-specific vocabulary can be incorporated to further improve the summarization performance.

Inspired by the above observations, we propose a model called Attribute-aware Sequence Network (ASN) to consider attribute information into review summarization. Specifically, ASN is based on sequence to sequence models (S2S), which are popular methods in text summarization (Rush et al., 2015; See et al., 2017; Zhu et al., 2018; Li et al., 2018a) and review summarization (Wang and Ling, 2016; Ma et al., 2018). ASN updates over standard S2S are three-fold. First, except for standard encoder and decoder in S2S, we design an attribute encoder, which encodes attribute preference for using words into attribute embedding. Second, an attribute-aware review encoder is proposed to generate attribute-aware review representation. It utilizes a bidirectional-LSTM to encode a review, and then imports an attribute-based selective mechanism to select important information of it to obtain a better review representation. Third, we propose an attribute-aware summary decoder to consider different writing styles of users with different attributes. It incorporates attribute embedding and attribute-specific vocabulary memory into word prediction module to generate summaries.

To validate our approach, we collect a review summarization dataset with user attribute information named *TripAtt* from TripAdvisor website, which contains 495,440 attribute-review-summary triplets with three kinds of attribute information: gender, age and travel status. Extensive experiments show that ASN achieves state-of-the-art performance on review summarization in both autometric ROUGE and human evaluation. Our contributions are as follows:

- To the best of our knowledge, we first propose an attribute-aware S2S-based model named Attribute-aware Sequence Network (ASN) to incorporate attribute information into review summarization.
- Our model adopts an attribute-based selective mechanism to consider different user preferences for review content and applies attribute-specific vocabulary to take the different writing styles of users with different attributes into consideration when generating a summary.
- For the evaluation of review summarization with attribute information, we collect a new dataset named *TripAtt*, which is available at https://github.com/Junjieli0704/ASN.

2 Dataset

Since there is no available review summarization dataset with user attribute information, we build a new one named *TripAtt* from TripAdvisor, an online hotel review site. TripAdvisor contains lots of user-generated reviews along with their authors and titles. The title of a review often summarizes the main idea of it; therefore, we take the title as the reference summary of the review. For attribute information, we can first get users' demographic information such as age and gender from the website. Second, users also explicitly label their travel status when booking hotels, such as traveled solo or traveled on business. Therefore, We take gender, age and travel status as our attribute information, and collect near 3 million attribute-review-summary triplets.

However, users may write titles arbitrarily, and it results in many meaningless titles, such as "i will be back again", and "twice in one trip". To remove these noisy samples, we apply filters proposed by Li et al. (2019). Then, we also remove samples that the value of any attributes (gender, age, or travel status) is NULL. Finally, we construct *TripAtt* with 495,440 attribute-review-summary triplets. We randomly split the dataset into 2,000 reviews for test, 2,000 reviews for develop and the rest for training. Table 1 shows statistics of *TripAtt*.

TripAtt	Train	Dev	Test
# Review	491,440	2,000	2,000
# Sens of Review	9.66	9.59	9.54
# Words of Review	173.90	172.04	171.86
# Sens of Summary	1.0	1.0	1.0
# Words of Summary	7.18	7.05	7.20

Table 1: Data statistics of *TripAtt* dataset.

2.1 Value distribution of different attributes

Figure 2 presents the value distribution of gender, age, and travel status in *TripAtt*. Males are more likely to travel than females, middle-aged (35-64 years old) users account for near 74%, and around 40% users traveled with couples.

2.2 Attribute-specific Vocabulary

Frequent words of a user can reflect the user well. Therefore, we want to mine attribute-specific words from *TripAtt* to model attributes. We first merge all summaries posted by users with the same group (such as male or 35-49 years old users) into a document. Then we compute tf-idf scores[3] for each word appears in the document, and we finally select top-N words for different groups. The last column in Figure 2 shows top-5 words in different attribute-specific vocabularies. We find that these words can actually reflect different groups well. For example, female users often utilize "lovely", "beautiful", "cute" to summarize review while these words rarely appear in summaries of male users. Users traveled with family often care about whether the hotel is suitable for their kids since they usually summarize reviews using "kids", "disney", and "parks", while business people frequently consider whether the hotel is suitable for meeting.

3 Attribute-aware Sequence Network

3.1 Problem Formulation

Suppose we have a corpus D with $|D|$ attribute-review-summary triplets, and each triplet contains a review $x = (x_1, x_2, ..., x_{|x|})$, a summary $y = (y_1, y_2, ..., y_{|y|})$ and an attribute vector $a = (a_1, a_2, ..., a_{|a|})$ which records $|a|$ kinds of attribute information of x's author. Since we have three kinds of attribute information (gender, age, and travel status), $|a|$ equals to 3. Classical review summarization is to generate y from x, while our goal needs to consider a's characteristics on summarizing reviews when generating y.

[3]Using *tf-idf* scores means we do not include too general terms that all users commonly use, because they do not help model the specific group.

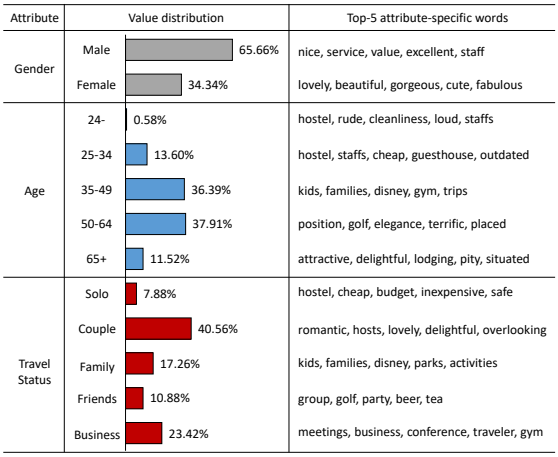

Figure 2: Value distribution and top-5 words in attribute-specific vocabulary of gender, age, and travel status in *TripAtt* dataset.

3.2 Model Framework

As shown in Figure 3, our model consists of three modules: attribute encoder, attribute-aware review encoder and attribute-aware summary decoder. Attribute encoder is based on attribute-specific words obtained from Section 2.2, which not only stores these words into attribute-specific vocabulary memory, but also utilizes multi-layer perceptron to merge them into an attribute embedding. Then we introduce four strategies to consider attribute embedding and attribute-specific vocabulary memory. Equipped with attribute embedding, our attribute-aware review encoder can select vital information from review representation. Importing attribute embedding and attribute-specific vocabulary memory into word prediction process of our attribute-aware summary decoder, our model can generate summary well.

3.3 Attribute Encoder

This module will produce attribute embedding and attribute-specific vocabulary memory. Suppose we select top-K attribute-specific words for each attribute in a, and then we concatenate these words to get attribute-specific vocabulary $A = (A_1, ..., A_K, ..., A_{2K}, ..., A_{|a|K})$, where word indexes between $(i-1) \times K + 1$ and $i \times K$ in A belong to attribute a_i. After that, we use an embedding matrix $\mathbf{E}_v$ to embed each word $\{A_i\}_{i=1}^{|a|K}$

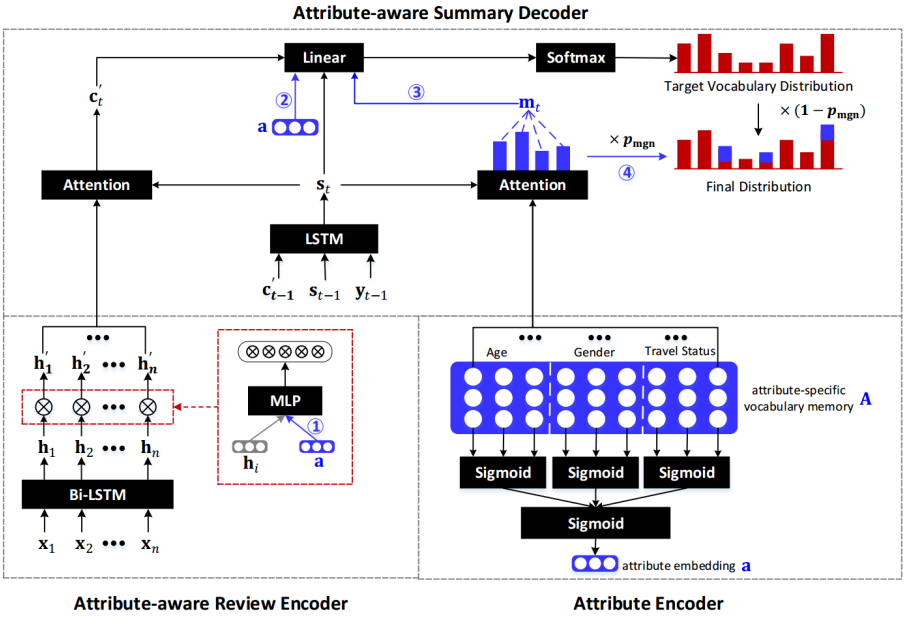

Figure 3: The architecture of Attribute-aware Sequence Network (ASN). ASN encodes two kinds of attribute information, attribute embedding (**a**) and attribute-specific vocabulary memory (**A**), into its two basic modules (*Attribute-aware Review Encoder* and *Attribute-aware Summary Decoder*). ①, and ② show strategies based on attribute embedding, and represent Attribute Selection strategy, and Attribute Prediction strategy, respectively. ③ and ④ indicate strategies based on attribute-specific vocabulary memory, and represent Attribute Memory Prediction strategy and Attribute Memory Generation strategy, respectively.

into vector $\{\mathbf{A}_i\}_{i=1}^{|a|K}$, and we get matrix $\mathbf{A}$, which is also called attribute-specific vocabulary memory. Then, we use a nonlinear layer to merge A's words belonging to attribute a_i into embedding $\mathbf{a}_i$ (See Equation (1)), which can only represent attribute a_i. After that, we merge these $|a|$ attribute embeddings $\mathbf{a}_1, \mathbf{a}_1, ..., \mathbf{a}_{|a|}$ into a nonlinear layer to get attribute embedding $\mathbf{a}$ (See Equation (2)), which is used to represent attribute vector a.

$$\mathbf{a}_i = \sigma(\mathbf{W}_a \mathbf{A}_{(i-1)\times K+1:i\times K} + \mathbf{b}_a) \quad (1)$$

$$\mathbf{a} = \sigma(\sum_{i=1}^{i=|a|} \mathbf{w}_a \mathbf{a}_i + b_a) \quad (2)$$

where $\mathbf{W}_a$, $\mathbf{w}_a$, $\mathbf{b}_a$, and b_a are learnable parameters, and σ denotes sigmoid function.

3.4 Attribute-aware Review Encoder

Given review x, it first embeds each word x_i into vector $\mathbf{x}_i$ using embedding matrix $\mathbf{E}_v$, which is the same embedding matrix in attribute encoder module. Then, these word vectors are fed into a single-layer bidirectional LSTM one by one, producing a sequence of encoder hidden states $\mathbf{h}_i$.

Classical review encoder utilizes $\mathbf{h}_i$ to represent review x and ends here. However, we find that users with different backgrounds pay attention to different content of a review. Inspired by (Zhou et al., 2017), we propose an attribute-based selective mechanism to select the important information from review for users with different attributes. The selective mechanism can construct a tailored representation of review x by considering a. In detail, our attribute-based selective network takes attribute vector $\mathbf{a}$ and the encoder hidden state $\mathbf{h}_i$ as input, and outputs a gate vector $\mathbf{gate}_i$ to select $\mathbf{h}_i$.

$$\mathbf{gate}_i = \sigma(\mathbf{W}_k[\mathbf{h}_i; \mathbf{a}] + \mathbf{b}_k) \quad (3)$$

$$\mathbf{h}'_i = \mathbf{h}_i \odot \mathbf{gate}_i \quad (4)$$

where $\mathbf{W}_k$, $\mathbf{b}_k$ are learnable parameters, $[;]$ is the concatenating operator, σ denotes sigmoid function, and $\odot$ is element-wise multiplication.

From Equation (4), we find $\mathbf{gate}_i$ is a vector whose value is between 0 and 1. A high value means most of the information in $\mathbf{h}_i$ passed from the filter, which results in the word $\mathbf{x}_i$ is important. This is the first strategy to consider attributes called ***Attribute Selection strategy***.

3.5 Attribute-aware Summary Decoder

At each decoding time step t, the decoder (a single-layer unidirectional LSTM) receives previous word embedding to obtain the new hidden state $\mathbf{s}_t$. Then it computes context vector $\mathbf{c}_t$

for time step t through the attention mechanism: where MLP stands for multi-layer perceptrons and $\alpha_{t,i}$ matches the importance score between current decoder state $\mathbf{s}_t$ and the encoder hidden state $\mathbf{h}_i^{'}$.

The classical summary decoder combines the context vector $\mathbf{c}_t$ and the decoder state $\mathbf{s}_t$, and then feeds the merged vector into a linear layer to produce the vocabulary distribution.

However, when generating a summary, users with different attributes may have their own vocabulary. Thus it is natural to take attribute-specific vocabulary memory $\mathbf{A}$ into consideration when predicting output vocabulary distribution and different words in $\mathbf{A}$ may have different effects. Thus, we utilize an attention mechanism to extract important words in $\mathbf{A}$ when obtaining vocabulary state $\mathbf{m}_t$.

$$\beta_{t,k} = \frac{\exp(\text{MLP}(\mathbf{A}_k, \mathbf{s}_t))}{\sum_k \exp(\text{MLP}(\mathbf{A}_k, \mathbf{s}_t))} \quad (5)$$

$$\mathbf{m}_t = \sum_k \beta_{t,k} \mathbf{A}_k \quad (6)$$

where $\beta_{t,k}$ measures the importance score between current decoder state $\mathbf{s}_t$ and the k-th word in attribute-aware vocabulary memory A_k.

Then we combine context vector $\mathbf{c}_t$, the decoder state $\mathbf{s}_t$, and $\mathbf{m}_t$ into the readout state $\mathbf{r}_t$. Besides, we can also enhance the readout state $\mathbf{r}_t$ by combining attribute vector $\mathbf{a}$. After that, we feed the readout state into a linear layer to produce the vocabulary distribution P_{voc}.

$$\mathbf{r}_t = \mathbf{W}_r[\mathbf{c}_t; \mathbf{s}_t; \mathbf{a}; \mathbf{m}_t] + \mathbf{b}_r \quad (7)$$

$$P_{\text{voc}} = \text{softmax}(\mathbf{W}_o \mathbf{r}_t + \mathbf{b}_o) \quad (8)$$

where $\mathbf{W}_r$ and $\mathbf{b}_r$ are learnable parameters. The strategies of adding attribute vector $\mathbf{a}$ and vocabulary state $\mathbf{m}_t$ into readout state $\mathbf{r}_t$ are called ***Attribute Prediction strategy*** and ***Attribute Memory Prediction strategy***, respectively.

Last but not least, inspired by (See et al., 2017), we also propose a soft copy mechanism to copy attribute-specific words in generating summaries, which is the 4-*th* strategy called ***Attribute Memory Generation strategy***.

The generation probability $p_{\text{mgn}} \in [0, 1]$ for timestep t is calculated from the context vector $\mathbf{c}_t$, the decoder state $\mathbf{s}_t$ and the vocabulary state $\mathbf{m}_t$:

$$p_{\text{mgn}} = \sigma(\mathbf{W}_{mg}[\mathbf{c}_t; \mathbf{s}_t; \mathbf{m}_t] + \mathbf{b}_{mg}) \quad (9)$$

where $\mathbf{W}_{mg}, \mathbf{b}_{mg}$ are learnable parameters, $[;]$ is the concatenating operator and σ is the sigmoid function. Next p_{mgn} is used as a soft switch to choose between generating a word from the target vocabulary V_t or coping a word from attribute-specific vocabulary.

$$P(w) = (1 - p_{\text{mgn}}) P_{\text{voc}}(w) + p_{\text{mgn}} \sum_{k:A_k=w} \beta_{t,k} \quad (10)$$

The first part in Equation (10) represents generating words from our vocabulary, and the second part indicates coping words from attribute-specific vocabulary memory, respectively.

3.6 Objective Function

Our goal is to maximize the output summary probability given the input sentence. Therefore, we optimize the negative log-likelihood loss function:

$$L = -\frac{1}{|D|} \sum_{(x,y) \in D} \log p(y|x) \quad (11)$$

4 Experiments

4.1 Evaluation Metric

We exploit ROUGE (Lin, 2004) as our evaluation metric. ROUGE scores reported in this paper are computed by Pyrouge package [4].

4.2 Comparison Methods

In the experiments, we compare our model with several strong baseline methods, which can be divided into two types: extractive and abstractive approaches. **LEAD1** is an extractive approach which selects the first sentence in review as summary. **LEXRANK** (Erkan and Radev, 2004) is also a famous extractive approach that computes text centrality based on PageRank algorithm. **TEXTRANK** (Mihalcea and Tarau, 2004) is an unsupervised algorithm based on weighted-graphs. **S2SATT** is a sequence to sequence model with attention implemented by us. **SEASS** (Zhou et al., 2017) employs a selective encoding model to control the information flow from encoder to decoder. **PGN** (See et al., 2017) copies words from the source text via pointing, while retaining the ability to produce novel words through the generator.

4.3 Implementation Details

Model Parameters The vocabulary is collected from the *TripAtt* training data. We lowercase the text, and there are 362,103 unique word types. We

[4] pypi.python.org/pypi/pyrouge/0.1.3

Models	RG-1	RG-2	RG-L
LEAD1	11.59	2.41	10.21
LEXRANK	8.80	1.19	7.87
TEXTRANK	10.47	1.71	9.30
S2SATT	20.61	5.96	18.95
SEASS	20.34	5.72	18.76
PGN	20.68	6.20	19.13
ASN	**21.52***	**6.61***	**19.74***

Table 2: ROUGE F1 scores (%) on the test set. RG in the Table denotes ROUGE. Models and baselines in the top half are extractive methods, while those in the bottom half are abstractive ones. The best performance is in **bold**. The superscript * indicates ASN performs significantly better than other models as given by the 95% confidence interval in the official ROUGE script.

use the top 30,000 words as the model vocabulary since they can cover 99.01% of the training data.

Model Training We set the batch size to 128. We truncate the review to 200 tokens, which is done to expedite training and testing. However we also find that truncating the review can raise the performance of the model[5]. We use the development set to choose the size of attribute-specific vocabulary and set it to 100. We also use Adam (Kingma and Ba, 2015), dropout (Srivastava et al., 2014) and gradient clipping (Pascanu et al., 2013) to make our model robust. We set the word embedding size to 128, and all LSTM hidden state sizes to 200. We use Adam as our optimizing algorithm. For the hyperparameters of Adam optimizer, we set the learning rate $\alpha = 0.001$, two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ respectively, and $\epsilon = 10^{-8}$. We use dropout with probability p = 0.2. We also apply gradient clipping with range $[-5, 5]$.

Model Testing At test time, We use beam search with a beam size of 4.

4.4 Results

Our results are given in Table 2. For extractive methods, we can see that LEAD1 performs best. However, it only obtains 11.59 ROUGE-1, 2.41 ROUGE-2, and 10.21 ROUGE-L F1 scores. The reason is that summaries in *TripAtt* are very succinct[6] and they often cover contents across several sentences, such as the summary in Figure 1.

For abstractive methods, we find that S2SATT is better than all extractive methods. After considering selective mechanism into S2SATT, the performance of SEASS decreases slightly. Because the selective mechanism proposed by SEASS is designed for sentence summarization, which may not be suitable for summarizing reviews. The average length of the input is less than 40 in Zhou et al. (2017), while the length in *TripAtt* is about 170. When incoporating copy mechanism into S2SATT, PGN obtains better performance.

Finally, after considering our proposed attribute encoder and four attribute-based strategies, ASN performs significantly better than all previous methods. Compared to S2SATT, our model has a 0.91 ROUGE-1, 0.65 ROUGE-2, and 0.79 ROUGE-L gains, which shows explicitly modeling attribute-related characteristics can indeed improve summarization quality. Our model also surpasses PGN by 0.84 ROUGE-1, 0.41 ROUGE-2, and 0.61 ROUGE-L and achieves the state-of-the-art performance on review summarization.

4.5 Human Evaluation on Aspect-level Coverage

Previous experiments show that ASN is better than other baselines when evaluating on ROUGE. However, ROUGE is only a word-based metric, which can not measure the semantic between two references. Table 3 illustrates an example. Since the summary generated by S2SATT contains more overlapping words with the gold summary than the one generated by ASN contains, S2SATT obtains higher ROUGE scores than ASN. However, from the view of aspects, ASN may be better. Because its summary describes ***location*** and ***service*** of a hotel, which are consistent with the gold result, while S2SATT's summary misses ***service***. Therefore, except for ROUGE, we want also to evaluate aspect-level coverage of different systems.

To perform the aspect-level evaluation, we first define seven aspects: ***location***, ***service***, ***room***, ***value***, ***facility***, ***food***, and ***hotel***, where ***hotel*** describes the overall attitude. Then, we randomly sample 1000 attribute-review-summary triplets from test set and generate summaries of these reviews using S2SATT, PGN and ASN. After that,

[5]Indeed, we found that using only the first 200 tokens of the review yields higher ROUGE scores than using all tokens.

[6]We observe in Table 1 that the average length of summary is about seven.

	A sample in test set	RG-1(%)	RG-2(%)	RG-L(%)	Aspect label
S2SATT:	good location , but small rooms	46.15	36.36	46.15	location, room
ASN:	good location , stuff needs improvements	33.33	20	33.33	location, service
Gold:	3+ good location , but some lacks in service	-	-	-	location, service

Table 3: A sample in test set shows high ROUGE may not result in better performance.

Models	Precision	Recall	F1
S2SATT	0.452	0.480	0.466
PGN	0.472	0.503	0.487
ASN	**0.491**	**0.530**	**0.510**

Table 4: Aspect-level Precision, Recall, and F1 scores (%) for different systems.

we ask two students to label aspects to these generated summaries and the reference summaries[7]. Finally, we compute aspect-level precision, recall, and F1 for different systems.

Table 4 shows the aspect-level result, and we find that ASN outperforms other models by a large margin, which shows summaries generated by our model can not only contain more correct words, but also in a higher consistency on aspects with references.

5 Discussions

In this section, we study the effect of different attributes, different attribute-specific strategies, and different attribute-specific vocabulary size on review summarization.

5.1 Effects of Different Attributes

To understand which kind of attribute information is the most important in review summarization, we perform an ablation study and give the results in Table 5.

First, all these kinds of attribute information are helpful for review summarization. Adding one kind of attribute information can obtain at least 0.41 ROUGE-1, 0.18 ROUGE-2 and 0.22 ROUGE-L gains. Second, travel status information is the most important attribute for review summarization in *TripAtt*. Because the travel status is a domain-dependent attribute, while others are domain-independent ones.

[7]Some examples about how to label aspects to summaries are shown in the last column of Table 3.

Attribute Information	RG-1	RG-2	RG-L
None	20.61	5.96	18.95
Age	21.04	6.19	19.39
Gender	20.98	6.14	19.17
Travel Status	21.09	6.22	19.41
Age + Gender	21.10	6.25	19.50
Age + Travel Status	21.21	6.46	19.63
Gender + Travel Status	21.08	6.33	19.48
ASN	21.52	6.61	19.74

Table 5: Ablation study on review summarization with attribute information.

line	ASel	APre	AMP	AMG	RG-1	RG-2	RG-L
1	–	–	–	–	20.61	5.96	18.95
2	✓	–	–	–	20.78	6.09	19.07
3	–	✓	–	–	21.03	6.15	19.21
4	–	–	✓	–	21.13	6.24	19.29
5	–	–	–	✓	21.08	6.19	19.24
6	–	✓	✓	✓	21.13	6.40	19.42
7	✓	–	✓	✓	21.52	6.49	19.70
8	✓	✓	–	✓	21.42	6.45	19.48
9	✓	✓	✓	–	21.15	6.38	19.37
10	✓	✓	✓	✓	21.52	6.61	19.74

Table 6: Effects of different attribute-based strategies on review summarization. "✓" means our model considers the specific strategy, while "–" means not. When there is no user-based strategies considered in our model, our model degrades into S2SATT (line 1).

5.2 Effects of different strategies

In this paper, we propose four attribute-based strategies to construct our attribute-aware review summarization model, which contains Attribute Selection strategy (ASel), Attribute Prediction strategy (APre), Attribute Memory Prediction strategy (AMP) and Attribute Memory Generation strategy (AMG). To evaluate the effect of each strategy on review summarization, we perform an ablation study and report results in Table 6.

First, we observe that models with only one kind of attribute-based strategy (line 2-5) can at least exceed S2SATT by (+0.17 ROUGE-1, +0.13 ROUGE-2, +0.12 ROUGE-L) points. It shows that all these strategies improve the performance of re-

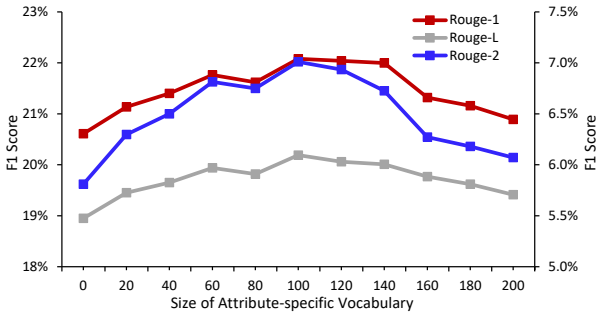

Figure 4: Effects of attribute-specific vocabulary size on review summarization on development set of *TripAtt*. When there is no any attribute-specific vocabulary (the size is 0) in ASN, our model degrades into S2SATT. The primary axis is for ROUGE-1 and ROUGE-L, and the second axis is for ROUGE-2.

view summarization. APre and AMG are the two most effective strategies, because they directly affect the word prediction module in ASN.

Second, models which deletes one kind of attribute-based strategy from ASN (line 6-9) will descend at least 0.11 in ROUGE-2 compared with ASN (line 10). It shows all our four user-based strategies are complementary. The most complementary strategies are ASel and AMG. The reason for ASel is that it is applied on the encoder module of ASN, while others are applied on the decoder module. For AMG, it differs from other decoder module strategies by affecting the decoding process through adding one more vocabulary distribution, while other strategies (APre and AMP) only add more features to classical word prediction module.

Third, ASN obtains the best result when considering all these strategies.

5.3 Effects of Attribute-specific Vocabulary Size

Since the attribute-specific vocabulary is the cornerstone for our attribute-aware model, we perform a test to detect the effect of its size on review summarization and show the result in Figure 4. First, we find that considering attribute-specific vocabulary can indeed improve the performance of review summarization, even though when the vocabulary size is very small (such as 20). Second, all curves increase firstly, and decrease with the increase of attribute-specific vocabulary size. It indicates that small vocabulary may be helpful for ASN, however large vocabulary may import much noise and be harmful to ASN. The best performance is obtained when the attribute-specific vocabulary size is 100, that's the reason why we set our attribute-vocabulary size to 100.

Review: i had the pleasure of staying at this hotel for a company ***meeting***. when you get to mco, call the hotel and they have a complimentary shuttle that will pick you up and bring you to the hotel. ... overall a great stay and if you need a place for your ***meeting*** , definately book here . . . . you wont be disappointed !
Attribute Information: 35-49, female, **traveled on business**

S2SAtt:	very nice hotel with great staff
PGN:	very nice hotel , great staff
ASN:	nice hotel for a ***meeting***
Gold:	great for a ***meeting***

(a) A sample shows the effect of travel state on review summarization

Review: estilo fashion hotel has n't been open very long but has set its standards high ! rooms are clean but not huge but everything is new and in good condition . staff are very helpful and efficient . our bedroom had a comfortable bed and was very modern . ...
Attribute Information: 50-64, **female**, traveled as a couple

S2SAtt:	modern hotel in great location
PGN:	great location , excellent service
ASN:	***lovely*** hotel and excellent service
Gold:	***lovely*** hotel with efficient staff !

(b) A sample shows the effect of gender on review summarization

Figure 5: Two samples in *TripAtt* show the effect of attributes on review summarization

5.4 Case Study

We present two cases from *TripAtt* that show the effect of attribute on review summarization in Figure 5.

Figure 5 (a) shows the effect of travel status on review summarization. Businessperson often books a hotel for working. In this case, the businesswoman stayed at the hotel for a company meeting. Whether the hotel is meet for the requirement may be the most important factor for her. She summarizes "great for a meeting". S2SATT could not get the point. Even though "meeting" appears in the review, PGN that has the effect of copying words from it also fails to generate the word. Our model containing businessperson-specific vocabulary can generate the word well and obtain a better summarization.

Figure 5 (b) shows the effect of gender on review summarization. Word-using habits in different genders are different. Female users often utilize "lovely", "beautiful", "cute" to summarize review while these words rarely appear in summaries from male users. Without considering the gender bias, S2SATT and PGN can not generate the summarization well. Incorporating such writ-

ing styles of female users in ASN, our model can generate "lovely" correctly, although it does not appear in the review.

6 Related Work

Review summarization belongs to sentiment analysis (Liu, 2016; Xia et al., 2015), which is a large area in natural language processing and contains sentiment classification (Li and Zong, 2008; Xia et al., 2011; Li et al., 2016, 2018b), emotion detection (Li et al., 2015), spam detection (Wang et al., 2017) and so on. There are two mainstream approaches for the problem: extractive and abstractive approaches.

A key task in extractive methods (Hu and Liu, 2004; Lerman et al., 2009; Xiong and Litman, 2014; Kunneman et al., 2018) is to identify important text units. For example, Hu and Liu (2004) first recognize the frequent product features and then attach extracted opinion sentences to the corresponding feature. Xiong and Litman (2014) exploit review helpfulness for review summarization. However, many studies (Carenini et al., 2013; Fabbrizio et al., 2014) have shown that abstractive approaches may be more appropriate for summarizing evaluative text than extractive ones. That is also the reason why we build our attribute-aware model based on abstractive methods.

Abstractive approaches (Ganesan, 2010; Gerani et al., 2014; Wang and Ling, 2016) are also very popular methods in review summarization. For example, Ganesan (2010) first represent review as token-based graphs based on the token order in the string and then rank summary candidates by scoring paths after removing redundant information from the graph. Gerani et al. (2014) utilize discourse structure of review to identify important aspects and then design a set of templates to generate summarizations. Wang and Ling (2016) propose an attention-based neural network model for generating abstractive summaries of opinionated text.

All of these studies focus on review summarization in the multiple review scenario, while our work focuses on the single review scenario. Recent review summarization studies (Ma et al., 2018; Yang et al., 2018a,b) also focus on the scenario. Ma et al. (2018) jointly models review summarization and sentiment classification in a unified framework. Yang et al. (2018a) study the aspect/sentiment-aware abstractive review summarization in an end-to-end manner. They mainly generate summary only based on review content and overlook the crucial influences of users. The most related work is Li et al. (2019). They study the personalized review summarization issue and also neglect the effect of user attributes on review summarization. Our proposed model fills this gap in the literature.

7 Conclusion and Future Work

In this paper, we propose an Attribute-aware Sequence Network (ASN) to consider attribute information into review summarization. ASN imports attribute-specific vocabulary to model attribute information and utilizes four attribute-based strategies to build attribute-aware review encoder and attribute-aware summary decoder. To validate our model, we construct a new dataset (*TripAtt*). Extensive experiments on *TripAtt* show that ASN achieves state-of-the-art performance on review summarization.

8 Acknowledgments

The research work descried in this paper has been supported by the National Key Research and Development Program of China under Grant No. 2017YFC0820700 and the Natural Science Foundation of China under Grant No. U1836221.

References

Giuseppe Carenini, Jackie Chi Kit Cheung, and Adam Pauls. 2013. Multi-document summarization of evaluative text. *Computational Intelligence*, 29(4):545576.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.*, 22:457–479.

Giuseppe Di Fabbrizio, Amanda Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. In *INLG*, pages 54–63.

Kavita Ganesan. 2010. Opinosis:a graph-based approach to abstractive summarization of highly redundant opinions. In *COLING*, pages 340–348.

Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-aware answer generation in e-commerce question-answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 429–437.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *EMNLP*, pages 1602–1613.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Florian Kunneman, Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2018. Aspect-based summarization of pros and cons in unstructured product reviews. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2219–2229.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan Mcdonald. 2009. Sentiment summarization: evaluating and learning user preferences. In *EACL*, pages 514–522.

Haoran Li, Junnan Zhu, Tianshang Liu, Jiajun Zhang, and Chengqing Zong. 2018a. Multi-modal sentence summarization with modality attention and image filtering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 4152–4158.

Junjie Li, Haoran Li, and Chengqing Zong. 2019. Towards personalized review summarization via user-aware sequence network. In *AAAI*.

Junjie Li, Haitong Yang, and Chengqing Zong. 2016. Sentiment classification of social media text considering user attributes. In *NLPCC*, pages 583–594.

Junjie Li, Haitong Yang, and Chengqing Zong. 2018b. Document-level multi-aspect sentiment classification by jointly modeling users, aspects, and overall ratings. In *COLING*, pages 925–936.

Shoushan Li, Lei Huang, Rong Wang, and Guodong Zhou. 2015. Sentence-level emotion classification with label and context dependence. In *ACL*, pages 1045–1053.

Shoushan Li and Chengqing Zong. 2008. Multi-domain sentiment classification. In *ACL*, pages 257–260.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Bing Liu. 2016. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, University of Cambridge.

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*, pages 4251–4257.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *ICML*, pages 1310–1318.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*, pages 379–389.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*, pages 1073–1083.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 47–57, San Diego, California. Association for Computational Linguistics.

Xuepeng Wang, Kang Liu, and Jun Zhao. 2017. Handling cold-start problem in review spam detection by jointly embedding texts and behaviors. In *ACL*, pages 366–376.

Rui Xia, Feng Xu, Chengqing Zong, Qianmu Li, Yong Qi, and Tao Li. 2015. Dual sentiment analysis: Considering two sides of one review. *IEEE Trans. Knowl. Data Eng.*, 27(8):2120–2133.

Rui Xia, Chengqing Zong, and Shoushan Li. 2011. Ensemble of feature sets and classification algorithms for sentiment classification. *Information Science*, 181(6):1138–1152.

Wenting Xiong and Diane Litman. 2014. Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews. *Grantee Submission*.

Min Yang, Qiang Qu, Ying Shen, Qiao Liu, Wei Zhao, and Jia Zhu. 2018a. Aspect and sentiment aware abstractive review summarization. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1110–1120.

Min Yang, Qiang Qu, Jia Zhu, Ying Shen, and Zhou Zhao. 2018b. Cross-domain aspect/sentiment-aware abstractive review summarization. In *Proceedings*

of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018, pages 1531–1534.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *ACL*, pages 1095–1104.

Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. 2018. MSMO: Multimodal summarization with multimodal output. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4154–4164.