Further Study on the Opinion Summarization Model PLANSUM

18019783 **21049846** **20133666** **21039204**

Abstract

Nowadays, there are more and more reviews of products or services on the Internet, and summarizing appropriate information from thousands of reviews is significant to both customers and businesses. In this project, we analyze and modify a recent unsupervised abstractive opinion summarization model, PLANSUM. After observing the dataset, we design a more realistic data partitioning strategy as well as a data pre-processing strategy. We present our encoder as a residual-connected stacked bidirectional LSTM. Experimental results show that our modified model achieves competitive results compared with the state-of-the-art models and rate higher than original PLANSUM in some values in terms of ROUGE.

1 Introduction

Opinion summarization is one of the most important fields in Natural Language Processing, especially at this time when the Internet has become a major shopping avenue. Opinion summarization is within the field of automatic summarization, which aims to extract the core information and generate a summary accordingly. This can be very useful in multiple aspects. For example, we can have a clear overview of a book and decide whether or not to read it just by looking at the abstract. In online shopping, review summarization saves users the trouble of reading all reviews and determining whether a product is good or bad, which is sometimes difficult and time-consuming. Additionally, this can be applied to other opinion-influenced domains such as movies, food, hotels, Tweets comments, blogs, etc.

For text summarization, there are two main approaches, namely abstractive and extractive approaches. While extractive summarization generates summaries that contain content extracted from the data, abstractive summarization uses only the extracted information rather than the exact content to generate the summaries. Since extractive summarization is often more rigid and contains more repetitive information, there are more and more studies on abstractive summarization today. Though abstractive summarization can generate novel sentences, it may go off track and produce unwanted information. In 2019, a generative model called 'Copycat-Review' (Bražinskas et al., 2019) was proposed to control the 'amount of novelty' going into a new review. In 2021, the 'Copycat-Review' model was beaten by the PLANSUM model (Amplayo et al., 2021) in multiple metrics, which incorporates content planning in a summarization model in order to not only create more natural synthetic datasets but also improve the output quality.

In our work, we modify the PLANSUM model in two ways. First, we modify the content plan induction model used to learn aspect and sentiment distribution, by increasing the depth and adding a residual connection to improve the informativeness and fluency of the model's generated summaries. Second, we modify the summarization model and add multi-head attention to the additive attention for more comprehensive feature extraction. We discuss the performance of the newly proposed model and the original one. Furthermore, we consider the performance in terms of data. We apply a data preprocessing procedure and split the original dataset that PLANSUM used in different categories to consider the effect of collision of multiple distributions as product reviews in different categories may have different aspects and sentiment and hence follow different distributions. Our results show that our approach performs better than the original PLANSUM and is competitive with most state-of-the-art extractive and abstractive models.

2 Related Work

In this section, we highlight the most closely related work on opinion summarization and content planning.

2.1 Opinion Summarization

Extractive summarization for opinion summarization has been investigated vigorously during the past few years and its general procedure is clustering the reviews, extracting the center of the review and combining them as the summary (Paul et al., 2010; Carenini et al., 2013; Angelidis et al., 2021). However, extractive methods usually suffer from the loss of key information and inconsistant utilization of information (Amplayo et al., 2021).

Abstractive summarization has recently benefited from vanilla attentional models (Rush et al., 2015; Nallapati et al., 2016). Pointer networks (Vinyals et al., 2015) have been widely used in the summarization task. Nallapati et al. (2016) proposed a RNN-based network including a pointer-generater to solve the out-of-vocabulary issue. Then See et al. (2017) also used the same method equipped with a coverage tracking technique to prevent duplication of input words. COPYCAT (Ive et al., 2019) also adopted pointer network with transformer architecture and it proposed extension for automatic post-editing.

Vector averaging is usually used to model opinion popularity. For example, MEANSUM (Chu and Liu, 2019) trains the decoder by reconstructing reviews and summaries are generated conditioned on average representation of inputs. The averaging technique is also adopted in the Bražinskas et al. (2020). It proposes to train a variational autoencoder with copy mechanism by reconstructing reviews and averages the reviews with the same entity.

Our study is based on PLANSUM (Amplayo et al., 2021) and this kind of models is to create synthetic datasets to simulate summaries and train supervised summarization models (Amplayo and Lapata, 2020; Bražinskas et al., 2019).

2.2 Content planning

Content planning performs well in the generation tasks with recent neural network based systems (Puduppully et al., 2019). But content planning usually suffers from discrete and domain-specific issues (Gehrmann et al., 2018; Moryossef et al., 2019). PLANSUM avoids these issues by taking the form of aspect and sentiment distributions.

3 Methods

3.1 PLANSUM

Our study is based on a novel unsupervised opinion summarization model PLANSUM (Amplayo et al., 2021). In this section, we will introduce the network structure and techniques used in the original model, which will help to understand the subsequent modifications.

The model consists of two parts. The first part is the condense phase, where we train the content plan induction model, that learns the aspect and sentiment distributions, and use it to create the synthetic dataset. The second part is the abstract phase, where we train the summarization model, that learns how to generate summaries using the synthetic dataset, and produce the summary outputs. The structure of models are showed in figures, including the modifications we made.

Step 1 Content Plan Induction

The content plan induction model aims to learn the encoding of x and induces probability distribution of aspect $p(a)$ and sentiment $p(s)$ to create synthetic datasets.

Given a review $x = \{w_1, ..., w_N\}$, the model obtains aspect and sentiment encodings h_a and h_s by encoding reviews using BiLSTM (Hochreiter and Schmidhuber, 1997) and mean pooling. Then by softmax operation of linear transformed encodings, we get the distributions $p(a)$ and $p(s)$.

$$\begin{aligned} \{h_i\} &= \text{BiLSTM}\left(\{w_i\}\right), \\ h_a, h_s &= \sum_i h_i / N. \\ p(a) &= \text{softmax}\left(W_a h_a + b_a\right), \\ p(s) &= \text{softmax}\left(W_s h_s + b_s\right). \end{aligned} \tag{1}$$

By following the work of (He et al., 2017), the aspect and sentiment embedding matrices $\mathbf{A}$ and $\mathbf{S}$ can be learned by reconstructing the reviews.

$$\begin{aligned} d_a &= \sum_i \mathbf{A}_i * p\left(a_i\right), \\ d_s &= \sum_i \mathbf{S}_i * p\left(s_i\right), \end{aligned} \tag{2}$$

$$\mathcal{L}_{\text{recon}} = \sum_i \max\left(0, 1 - d_a h_a + d_a n_a^{(i)}\right) + \sum_i \max\left(0, 1 - d_s h_s + d_s n_s^{(i)}\right), \quad (3)$$
$$\mathcal{R}_{\text{recon}} = \left\|\mathbf{A}\mathbf{A}^\top - \mathbf{I}\right\| + \left\|\mathbf{S}\mathbf{S}^\top - \mathbf{I}\right\|.$$

A contrastive max-margin loss function is used to reconstruct original encodings h_a, h_s with d_a, d_s. For each review, m negative samples are generated and considered in the loss function with their encodings $\{n_a^{(i)}, n_s^{(i)}\}$. To encourage the uniqueness, $\mathcal{R}_{\text{recon}}$ is used as a regularization term.

Another consideration is about the information overlapping. To ensure the aspect embedding matrix do not contain the sentiment information, the model leverage review ratings $\hat{s}$ and an adversarial classifier with a reverse gradient function (Ganin et al., 2016) to remove sentiment information from aspect embedding $\mathbf{A}$. The following disentanglement loss uses cross-entropy objective function.

$$p(s)_{adv} = \text{softmax}\left(\text{GradRev}\left(W_{adv} h_a + b_{adv}\right)\right),$$
$$\mathcal{L}_{\text{disen}} = -\log p(\hat{s}) - \log p(\hat{s})_{adv}. \quad (4)$$

The overall training loss of content plan induction model is a linear combination of above losses with a hyperparameter λ controls the regularization.

$$\mathcal{L}_{\text{induce}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{disen}} + \lambda \mathcal{R}_{\text{recon}}. \quad (5)$$

Step 2 Create Synthetic Dataset

This step we aim to create a synthetic dataset $\mathbb{D} = \{\mathbf{X}, \mathbf{y}\}$ and will use it to train the summarization model in the next step. We first select a review y and obtain its aspect and sentiment distribution $p(a), p(s)$. By using Dirichlet distribution, we sample N pairs of distributions $(p_i(a), p_i(s))$ where $i \in 1, ..., N$, and run a nearest neighbour search over the corpus to match the most similar review $\boldsymbol{x}_i$.

$$p_i(a) \sim \text{Dirichlet}\left(\alpha_a * p(a)\right),$$
$$p_i(s) \sim \text{Dirichlet}\left(\alpha_s * p(s)\right), \quad (6)$$

where α_a, α_s controls the variance of the Dirichlet distributions.

Step 3 Opinion Summarization Model

The summarization model reuses the encodings from Step 1. Then to mitigate the redundancy, the author borrows an effective fusion method from Xu et al. (2018) that uses an injective function:

$$h_k = \text{MLP}\left(e_k + \Sigma_{(i,j):w_j^{(i)}=w_k} h_j^{(i)}\right), \quad (7)$$

where e_k is a learned embedding for word w_k in the vocabulary.

The decoder is an LSTM with additive attention (Bahdanau et al., 2014) and copy (Vinyals et al., 2015), where aggregated token embeddings $\{h_k\}$ are used as keys. The aspect and sentiment encodings d_a, d_s in the content plan model are combined with output token y_t as the input of LSTM and attention.

$$y'_t = f\left(d_a, d_s, y_t\right),$$
$$s_t = \text{LSTM}\left(y'_t, s_t\right),$$
$$p\left(y_{t+1}\right) = \text{ATTENDCOPY}\left(y'_t, s_t, \{h_k\}\right), \quad (8)$$

where $f(\cdot)$ is a linear function.

The objective function of summarization model is a maximum likelihood loss based on summary $\boldsymbol{y} = \{y_t\}$ and the model also use a LM-based label smoothing method (Szegedy et al., 2016), which uses predictions from BERT (Devlin et al., 2018).

$$\hat{y}_t = (1 - \delta) * y_t + \delta * \text{BERT}\left(y_{-t}\right),$$
$$\mathcal{L}_{gen} = -\sum_t \hat{y}_t \log p(y_t). \quad (9)$$

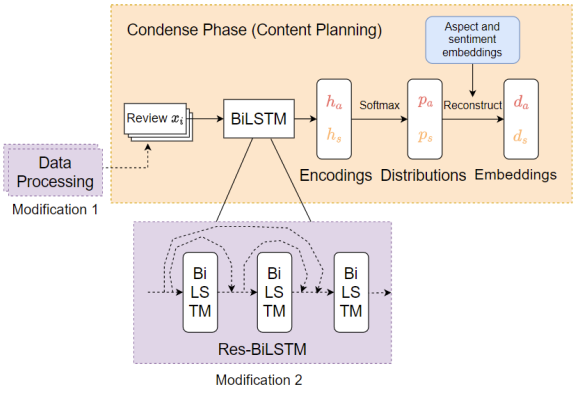

Figure 1: The structure of Condense Model

3.2 Modified PLANSUM: RES-PLANSUM

The dataset used by PLANSUM to learn how to summarize is the synthetic dataset generated by the learned distribution of aspect and sentiment in the previous content plan induction model. Therefore, the sythetic dataset is significant for subsequent

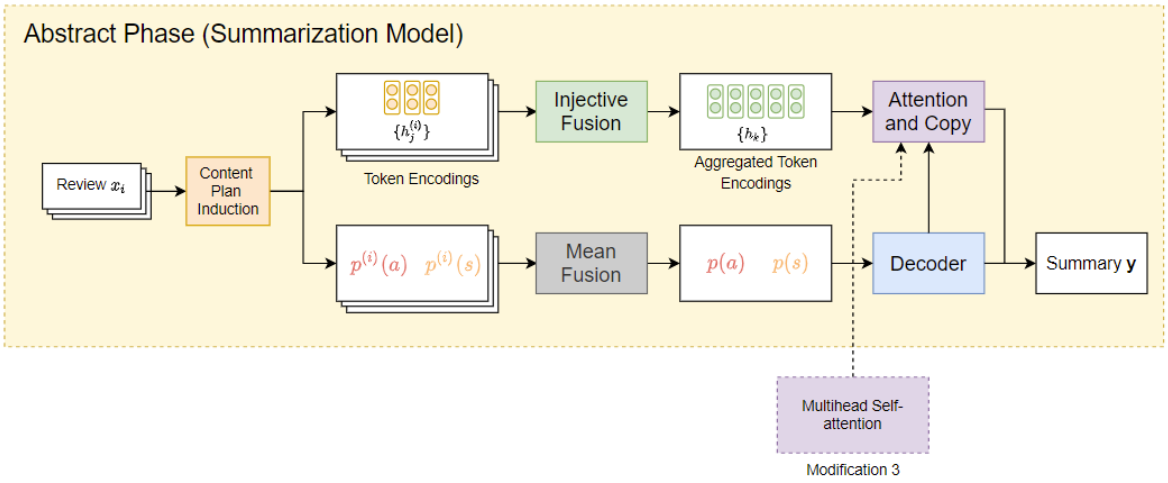

Figure 2: The structure of Abstract Model

training. In order to create more complex feature representation, on the basis of bidirectional LSTM (BiLSTM) we use a deep BiLSTM to increase the model complexity and wish to benefit from the depth in space. This deep RNN structure is first introduced in Graves et al. (2013). In this stacked architecture, the output sequence of the current BiLSTM will be fed as the input sequence to the next BiLSTM.

We further improve this model by applying the idea of ResNet in computer vision (He et al., 2016). We add residual connections between the stacked BiLSTM such that the input sequence of each BiLSTM is the summation of the original word embedding and all output sequences of previous BiLSTMs. We hypothesize that such residual connection can preserve the features of the original text without adding additional parameters. The modified model structure is illustrated in Figure 1.

3.3 Modified Summarization model

From the original summarization model, we found that the additive attention used here is a bit simple. So we argue that some novel and complex structures can be adopted to help model learn the relationship between reviews and summaries and improve its summarization ability.

Inspired by the work of self-attention (Vaswani et al., 2017) and its application in abstractive summarization (Paulus et al., 2017), we consider incorporating multi-head self-attention into the attention structure of summarization model. And we also want to see the difference between additive attention and multiplicative attention. In this modification, we only change the summarization model and previous steps are the same as the original model.

3.4 Data Imbalance

Different types of products and services have their own aspect and sentiment spaces. Some of these spaces overlap, but if the number of reviews of a certain product or service is much more than others, the model will be more inclined to the aspect and sentiment distribution of that specific product or service, which will affect the generalization ability of the model. One of the factors that affects the distribution is the amount of data. We assume that the reviews of the same type of product or service are independent and identically distributed, but the reviews of different types of products or services do not follow the same distribution. Therefore, in order to prevent the aspects and sentiments in the summary from tending to a specific distribution, we design two experiments. The first experiment still uses the full four-group dataset but the number of reviews per group is controlled to be the same so the total number of data is reduced, but the model performance is expected to be better, while the second is to divide the dataset into four categories according to product types, and check the model performance on each grouped data.

3.5 Data Pre-processing

Customers' reviews of a product or service are subjective, and these reviews tend to be poorly spelled and indented. Such reviews humans have no obstacle to reading because they have rich prior knowledge, but they are not friendly to tokeniza-

tion. Incorrect indentation, spelling, and punctuation can cause the pretrained model to encode it into the wrong vector, which in turn affects subsequent decoding process and training performance. To reduce this harm, we pre-processed the data by:

- We changed all abbreviations to full letters. E.g. "I'm" → "I am".
- We have removed punctuation that does not affect the meaning of the sentence, such as periods, commas, and underscores. Symbols such as $, /, etc. are reserved.
- We have fixed some obvious and common whitespace errors, such as changing from "do n't" to "do not".

4 Experiments setup

4.1 Datasets

We conduct experiments on Amazon opinion summerization benchmark, which includes product reviews for four Amazon categories (i.e., Electronics, Clothing, Shoes and Jewelry, Home and Kitchen, and Health and Personal Care)(Bražinskas et al., 2020). The development and test sets include reviews and three gold-standard summaries generated by Amazon Mechanical Turk (AMT) crowdworkers.

4.2 Training Configuration

Most hyperparameters are inherited from (Amplayo et al., 2021). We have an additional parameter to set the size of the development set. Our model is trained separately on a single GeForce GTX 2060 GPU and on Google Colab's free NVIDIA Tesla K80 GPU, and is implemented using Pytorch[1].

4.3 Compared methods

The focus of this work is to improve the performance of PLANSUM, therefore we do not compare our model with other state-of-the-art methods, but only with the original PLANSUM. We compare three models, PLANSUM, DEEP-PLANSUM, and RES-PLANSUM in terms of classical ROUGE-scores (Lin and Hovy, 2003).

5 Results & Discussion

Specifically, we use the ROUGE F_1 score(Lin, 2004) as a measure of the quality of opinion summaries. Unigram and bigram overlap (ROUGE-1 and ROUGE-2) are used to evaluate the informativeness, while the longest common subsequence (ROUGE-L) is used as a measure of fluency.

[1]The code is changed on the basis of PLANSUM

Input
1. ... however , it fit me a bit snug to the front probable because my feet are a bit broader and you do have to order the size a 1 / 2 inch bigger than you normally wear . 2. ... the exact same shoe for much less . the quality is the same ... and it fits perfectly. 3. ... so did not notice any comfort issues ... my feet felt like they had been pinched around the toe area. ... 6. ... they matched my navy satin dress perfectly and were super comfortable . i wore these shoes longer than i have worn any heels... 7. these shoes run appropriate to size (i 'm a size 9) , but i felt like they pinched my toes as i have a wider foot ... but i sent them back as i did n't find them comfortable 8. ... and needed a comfortable shoe ... and these completely fit the bill . i would buy again
Opinion Summary
this is a great size for the price it is very comfortable and i love it the size is perfect for walking around the house and it looks like it should be the perfect size i would recommend this to anyone who is looking for a good quality

Table 1: Amazon reviews about a pair of shoes and corresponding summary generated by RES-PLANSUM model. Aspect-specific opinions are in color, e.g. size, comfort, quality.

5.1 Effect of Data Pre-processing

The results of model performance using pre-processed and raw data are shown in table 2. Overall, models using pre-processed data to train are rated higher in terms of R2 and RL, while RES-PLANSUM is rated the best in all criteria. PLANSUM and DEEP-PLANSUM have slight difference in R1, R2 and RL. We observe a large improvement of 5.61, 2.03 and 3.74 points in R1, R2 and RL, respectively, using pre-processed data compared to RES-PLANSUM trained on raw data. This im-

provement reveals that under a suitable pre-process which preserves the informativeness and sentence structure, residual connections can preserve correctly encoded aspect and sentiment features in deep RNN architecture.

Model	R1	R2	RL
PLANSUM	**31.30**	5.29	18.08
*PLANSUM	29.32	**5.36**	**18.14**
DEEP-PLANSUM	**27.52**	3.78	16.21
*DEEP-PLANSUM	27.34	**4.48**	**16.76**
RES-PLANSUM	24.44	3.62	15.13
*RES-PLANSUM	**30.05**	**5.65**	**18.87**

Table 2: An asterisk on the model name indicates that the model is trained on the pre-processed data.

5.2 Model Performance

We compare three models on the full Amazon dataset. The results are shown in table 3. The performance of the two models proposed is inferior to that of the baseline model. There are two possible reasons for this: data imbalance and mixture of orthogonal aspects. The first reason is our data imbalance hypothesis, as mentioned in Section 3.4. Unbalanced dataset leads to distorted aspect distribution learned by the content plan induction model. This distortion is amplified when increasing the model depth and through residual connections. As a result, the training loss converges much slower than in baseline PLANSUM and only converges to local minima in DEEP-PLANSUM and RES-PLANSUM. The second reason is from a practical point of view. Products from different categories usually have aspects that do not overlap, that is, orthogonal aspects. The way that PLANSUM used to generate aspect distribution is to sum an entire review, which contains many other non-aspect word embeddings, and these are the shortcut features (superficial but unhelpful features)(Geirhos et al., 2020) learned by the model, which may be improved by using feature entanglement(Ganin and Lempitsky, 2015). In table 1 we presents a summary generated by RES-PLANSUM.

Model	R1	R2	RL
PLANSUM	**31.30**	**5.29**	**18.08**
DEEP-PLANSUM	27.52	3.78	16.21
RES-PLANSUM	24.44	3.62	15.13

Table 3: Models trained on full Amazon Dataset.

Model	R1	R2	RL
PLANSUM	**31.30**	**5.29**	**18.08**
Δ PLANSUM	23.92	4.12	13.43

Table 4: ΔPLANSUM refers to PLANSUM trained with balanced data.

5.3 Effect of Data Balancing

We compare the PLANSUM model using the raw and balanced data to try to tackle the first reason mentioned in the previous section.

According to the results given in table 4, using balanced data does not improve the model performance. One possible reason is that our dataset is reduced and important information may be lost due to random sampling. In the balanced dataset, we have 5500 summaries under each category as the smallest category (Amazon health) only consists of this much data. Since our original dataset is imbalanced, our change in the dataset results in many abandoned reviews which may contain important aspects that could be helpful to improve the model.

According to our assumption, data in different categories have different distributions of aspects and sentiments. In our model, aspects and sentiments obey Dirichlet distribution with parameter $\alpha_a p(a)$ and $\alpha_s p(s)$ (α_a and α_s are constant vectors and $p(a)$ and $p(s)$ are aspects distribution and sentiments distribution learned from data). Hence, in any dataset that contains data from multiple categories, the distribution we have is actually the combination of all the distributions. When the amount of data is different, the category containing more data might contribute more to the combined distribution. When we use balanced data, parameters are averaged. This could cause our model to lose important aspects and the averaged result may not be able to represent any significance. In this way, we are left with many insignificant aspects. The lengthy summaries generated by models with balanced data are a good verification of this (In fact, similar ideas appear in pooling; when using average pooling, the most significant information will be lost and the averaged information will be presented.).

5.4 Effect of Data Partitioning

To investigate the aspect distribution based on category, we perform experiments on category-based partitioned data and obtain the following results shown in table 5. Among four datasets, we ob-

Model	home			health		
	R1	R2	RL	R1	R2	RL
PLANSUM	27.59	5.39	15.62	26.34	**5.21**	15.51
DEEP-PLANSUM	26.67	4.67	**15.94**	26.75	4.53	14.77
RES-PLANSUM	**29.10**	**5.56**	15.74	**27.52**	4.88	**15.75**
Model	electronics			cloth		
	R1	R2	RL	R1	R2	RL
PLANSUM	25.55	**5.39**	14.42	25.65	**4.93**	14.29
DEEP-PLANSUM	24.95	4.59	14.10	26.77	4.40	14.80
RES-PLANSUM	**30.38**	4.32	**16.74**	**28.20**	4.84	**15.55**

Table 5: The highest scores are indicated in bold, the second highest scores are underlined.

serve that RES-PLANSUM performs better in R1 and RL metrics. For home dataset, RES-PLANSUM performs better in R2 but slightly worse in RL.

R1 reflects to a certain extent the number of aspects contained in the summary, while R2 also takes the order of words into consideration. RES-PLANSUM is rated high in both R1 and RL indicates that it learns more aspects than the other two models and it generates more fluent summaries, although the words ordered not as accurately as the other two, which is why it has a relatively low R2 score.

DEEP-PLANSUM performs the worst in general with exceptions. It generates the most fluent summaries in home dataset which its R2 scores is approximately 0.65 lower in all datasets. It may be due to the fact that the deep structure amplifies some non-aspect features, causing distortion in aspect distribution so the generated texts are not in a well-ordered manner.

5.5 Modified Summarization Model Performance

Model	R1	R2	RL
Add Att (Original)	31.30	5.29	18.08
Dot Att	**31.58**	**5.52**	**18.10**
Self-att Drop-0.1	29.68	5.01	17.30
Self-att Drop-0.25	24.54	2.85	14.66

Table 6: Performance for different modifications in the attention structure

We can see from table 6 that our modification does not perform well compared with the original one. Only the multiplicative attention improves a little bit and other modifications' performances are quite poor. We have also tried to train the multi-head self-attention without dropout and higher number of heads models. It seems that these models cannot get readable results, so we choose not to display them here.

Since our modifications perform poorly, we do not attempt to combine the modifications with the previous models and data. In our opinion, the issues may be related to the various techniques used in the summarizations. The injection method and content planning may have effects on the extra self-attention structure. Furthermore, according to the transformer structure (Vaswani et al., 2017), it is more reasonable if we could combine both encoder and decoder with multi-head self-attention. Due to the limited time and computational resources, we leave this as future work.

5.6 Metrics with after-processed summaries

In terms of results, even the highest ROUGE score does not provide a satisfactory value. After manually comparing the gold summaries and predicted summaries, we found that most of the predicted summaries have almost the same meaning though they still have low metrics. We think it is because of the diversity of expression. For example, the expressions "this shoe is very classy and chic" and " the shoes are so pretty" have similar meanings – the shoe (aspect) is good (sentiment).

If the summaries generated by our models contain enough aspects and right sentiments, we consider these summaries are good enough. However, people are rich in expressions of "good" or "bad", which means people usually use different adjectives to express the same sentiments and it will lead to a low grade of ROUGE metrics. Therefore we need to do some synonymous substitutions, a trade-off between aspects informativeness and key

significance.

We use corpus of positive words and negative words, and simply replace those adjectives with "good" and "bad" respectively in both the gold and predicted summaries. After this after-processing, we use ROUGE metrics to evaluate the performance of our models and a great improvement is shown. Though it does not mean that we have improved our model further, this after-processing can help us to determine the quality of our models as it gives direct feedback on two things we pursue in the model – aspects and sentiments. We acknowledge the precision loss during this process as we no longer evaluate the exact words. However this after-processing can be viewed as a method of filtering writing style and by only keeping the aspect and sentiment orientation in the results, we can compare more effectively among the results and understand the effect of our modifications better. Thus we can combine after-processing and ROUGE metric and introduce a better approach to evaluate predicted summaries.

6 Conclusion

In this work we propose two modifications on the PLANSUM model, one is on the content plan induction model, which is used to generate aspect and sentiment distributions, and another is on the summarization model, which is used to generate summary from a batch of reviews. In the former modification, we propose and investigate the performance of two models, DEEP-PLANSUM and RES-PLANSUM. In the later one we add multi-head attention into additive attention. Our results show that DEEP-PLANSUM and addition of multi-head attention have very limited effects on the model performance in terms of ROUGE, while RES-PLANSUM indicates a high increase in ROUGE values when using pre-processed dataset, as well as on the categorical dataset. Our work also shows the limitation of content plan induction model in learning the aspect distribution, which reveals the conflict between aspect distributions as well as the shortcuts learned by the model. Further research could consider how to filter aspects from short or long texts more effectively by methods of removing or downweighting shortcut features.

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Unsupervised opinion summarization with content planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12489–12497.

Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised opinion summarization with noising and denoising. *CoRR*, abs/2004.10150.

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2019. Unsupervised opinion summarization as copycat-review generation. *arXiv preprint arXiv:1911.02247*.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Unsupervised multi-document opinion summarization as copycat-review generation. In *ACL*.

Giuseppe Carenini, Jackie Chi Kit Cheung, and Adam Pauls. 2013. Multi-document summarization of evaluative text. *Computational Intelligence*, 29(4):545–576.

Eric Chu and Peter Liu. 2019. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

Sebastian Gehrmann, Falcon Z Dai, Henry Elder, and Alexander M Rush. 2018. End-to-end content and plan selection for data-to-text generation. *arXiv preprint arXiv:1810.04700*.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Julia Ive, Pranava Swaroop Madhyastha, and Lucia Specia. 2019. Deep copycat networks for text-to-text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3227–3236.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*, pages 150–157.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. *arXiv preprint arXiv:1904.03396*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Michael Paul, ChengXiang Zhai, and Roxana Girju. 2010. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 66–76.

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

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6908–6915.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.