Learning Cluster Patterns for Abstractive Summarization

SUNG-GUK JO1, SEUNG-HYEOK PARK1, JEONG-JAE KIM2, AND BYUNG-WON ON1

1School of Software, Kunsan National University, Gunsan-si, Jeollabuk-do 54150, South Korea
2Graduate School of Cognitive Science, Yonsei University, Seoul 03722, South Korea

Corresponding author: Byung-Won On (bwon@kunsan.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government through MSIT under Grant NRF-2022R1A2C1011404.

ABSTRACT Nowadays, pre-trained sequence-to-sequence models such as PEGASUS and BART have shown state-of-the-art results in abstractive summarization. In these models, during fine-tuning, the encoder transforms sentences to context vectors in the latent space and the decoder learns the summary generation task based on the context vectors. In our approach, we consider two clusters of salient and non-salient context vectors, using which the decoder can attend more over salient context vectors for summary generation. For this, we propose a novel cluster generator layer between the encoder and the decoder, which first generates two clusters of salient and non-salient vectors, and then normalizes and shrinks the clusters to make them apart in the latent space. Our experimental results show that the proposed model outperforms the state-of-the-art models such as BART and PEGASUS by learning these distinct cluster patterns, improving up to 2∼30% in ROUGE and 0.1∼0.8% in BERTScore in CNN/DailyMail and XSUM data sets.

INDEX TERMS Abstractive summarization, cluster, contextualized vector, BART, PEGASUS.

I. INTRODUCTION

As one of the challenging problems of natural language processing, various methods for automatically summarizing text documents have been actively studied so far. Text summarization are mostly divided into extractive and abstractive approaches. In the extractive approach, a summary is generated by extracting and combining a few important sentences from the original text. On the other hand, the abstractive approach reproduces a short text containing important information, rather than taking part of the original text as it is.

Since abstractive summarization is a relatively more difficult task than extractive summarization because it must understand the meaning of the original text as well as the natural generation of sentences, most studies have focused on extractive summarization solutions. However, recently, starting with the abstractive summarization model using the sequence-to-sequence architecture [1], [2], transformer-based pre-trained models such as Bidirectional Encoder Rep-
FIGURE 1. Main concept of the proposed model. In this figure, suppose that the source text is composed of five sentences. The red and blue points indicate salient and non-salient context vectors from the encoder in BART, respectively. The cluster pattern generation layer transforms a salient context vector $\vec{v}_i$ to another context vector $\vec{v}_i'$ through $\vec{v}_i' = w \odot \vec{v}_i$, where $w$ is a certain weight value and $\odot$ is Hadamard product.

the teacher-forcing language model objective, based on the context vectors.

Because a well-written abstractive summary is a short text that is rephrased by a few salient sentences in the original text, handling most non-salient sentences is critical in abstractive summarization. In this work, we view the noise information to non-salient sentences in the original text. For effective summary generation, if the decoder can take a look at cluster information e.g., two clusters of salient and non-salient context vectors during fine-tuning, it can attend more over salient context vectors rather than non-salient ones.

As shown Figure 1(a), all sentences composing a text document are relevant with each other regardless of their importance so the corresponding context vectors are closely located to each other in the latent semantic space. This mixture of salient and non-salient context vectors makes it difficult to generate a correct summary. In our approach, we view abstractive summarization to the problem of finding a few salient sentences in a document and then concisely rephrasing them while minimizing the impact of non-salient sentences. Specifically, we first assign each sentence in a document to either salient class or non-salient one, and then force the context vectors corresponding to the salient sentences, away from the non-salient ones. As a result, two clusters of salient and non-salient context vectors are formed in the latent space. Such clusters are far apart from each other so noise context vectors are less affected for summary generation. For instance, Figure 1(b) illustrates the result of two clusters formed by the proposed method in which the discriminator identifies a few salient sentences per document and the cluster generator artificially creates two clusters of salient and non-salient context vectors.

To take advantage of this cluster information, we propose a novel method of learning cluster patterns of context vectors, based on BART, the best one among pre-trained Transformer sequence-to-sequence models in recent times. We call our method ClusterBART (CBART) throughout this article. We also propose (1) cluster normalization using which the decoder learns cluster patterns more robustly and (2) cluster shrinking that reduces the variance between vectors in a cluster and maximizes the margin between two clusters. In particular, note that our method only transforms existing context vector representation $\vec{v}_i$ to new vector representation $\vec{v}_i'$. As a result, binary clusters are formed by these new vectors. To the best of our knowledge, our work is the first study to use cluster patterns of context vectors for summary generation. In XSUM and CNN/DailyMail data sets, the experimental results show that the proposed model improves up to 5% in ROUGE-1/2/L/avg and 0.4% in BERTScore, compared to BART and up to 30% in ROUGE-1/L/avg and 0.8% in BERTScore, compared to PEGASUS.

The remainder of this article is organized as follows. In Section II, we introduce existing abstractive summarization methods related to this work. In particular, we discuss the novelty of our method and the main difference between previous studies and our work. In Section III, we describe the details of the proposed method based on the discriminator and the cluster generator. Next, we explain the experimental set-up in Section IV and discuss the experimental results in detail in Section V. Finally, we summarize our current work followed by the future research direction in Section VI.

1This is the extended work of [5], where we have just proposed to create two clusters of encoding vectors to improve abstractive summarization in the CNN/DM dataset. However, in this article, we newly propose cluster normalization and shrinking methods as well as the existing cluster generation method, and show the effectiveness of the newly proposed methods in the XSum dataset in addition to the CNN/DM dataset. The experimental results show that CBART outperforms both BART and PEGASUS, the state-of-the models in abstractive summarization. In addition, we show the best hyper-parameter values needed in CBART.
II. RELATED WORK

For abstractive summarization, [1] and [2] first proposed the encoder-decoder attention model. Reference [6] improved the neural model of [2] by copying words from an input text through the pointer network, which can handle out-of-vocabulary words, while generating words from the generator network. They also proposed a coverage mechanism that tracks and controls the coverage of input text in order to eliminate repetition.

With the great success of the transformer model in natural language generation, [7] and [8] proposed pre-trained models for abstractive summarization. In particular, [7] proposed an unified pre-trained language model (UniLM) that is a multi-layer transformer model as backbone network, jointly pre-trained by various unsupervised language modeling objectives such as (1) bidirectional language model, (2) right-to-left or left-to-right unidirectional language model, and sequence-to-sequence language model, sharing the same parameters.

Reference [8] used an encoder-decoder model, in which a document-level encoder using BERT is pre-trained on large amounts of text. They also proposed a new fine-tuning schedule to alleviate the mismatch between the encoder and the decoder with BERTSUMABS and BERTSUMEXtabs, which are the baseline and the two-stage fine-tuned models. Recently, [9] proposed the BART model that consists of BERT as the encoder and GPT as the decoder. The BART model is similar to BERTSUM proposed in [8], except that the encoder can perform the masking task through various denoising functions.

As one of the state-of-the-art Transformer encoder-decoder models, PEGASUS uses a new self-supervised pre-training objective named as Gap Sentences Generation (GSG). Unlike BERTSUM and BART, PEGASUS masks an important sentence in a document rather than some token [10]. In fact, this is an extension of the word embedding method to understand the meaning of sentences. For example, in the document $d = \{s_1, s_2, s_3, s_4\}$, where $s_i$ is the $i$-th sentence in $d$. Assuming that $s_2$ is the most important sentence, $s_2$ is first masked as $s'_2$. Then, any token chosen at random from $s_1$ surrounding $s_2$ is masked. Similarly, any token chosen at random from $s_3$ surrounding $s_2$ is masked. These masked tokens and $s'_2$ are used as the input of the Transformer encoder. On the other hand, $s_2$ is used as the input of the Transformer decoder. In the pre-training step, PEGASUS is learned through the process where the encoder predicts the masked tokens and the decoder generates $s_2$ correctly.

Besides these pre-trained models, various models using reinforcement learning [11], [12], topic model [13], [14], multimodal information [15], attention head masking [16], information theory [17], extraction-and-paraphrasing [18], entity aggregation [19], factuality consistency [20], [21], [22], [23], [24], [25], deep communicating agents [26], sentence correspondence [27], graph [28], [29], [30], and bottom-up approach [31] were proposed for abstractive summarization. Because these models are not directly related to our proposed model, we do not compare with them in this article.

Recently, there have been a few studies utilizing clusters in opinion summarization [32], [33], [34], which is quite different from abstractive summarization. The purpose of opinion summarization is to generate an aspect-based summary that represents opinion popularity well in an unsupervised manner. For opinion summarization, existing methods discover clusters hidden from a collection of customer reviews, using Variational AutoEncoder (VAE), Expectation-Maximization (EM), or supervised aspect classifiers, and then generate aspect-based summaries by averaging the embedding vectors belonging to each cluster. However, our proposed method is different from them. We on purpose create two clusters of salient and non-salient vectors in the semantic latent space, where the cluster of salient context vectors is far from the cluster of non-salient ones, so that the decoder can learn to clearly distinguish between salient and non-salient context vectors for abstractive summarization. In particular, our work on learning artificial cluster patterns in latent semantic space is the first study in abstractive summarization.

III. MAIN PROPOSAL

A. BACKGROUND: BART

Abstractive Summarization is based on a language model such as GPT, which learns in the process of predicting the next word with a given word sequence. In this case, words are predicted conditioned on only left-wise context so that it cannot learn in both left-wise and right-wise directions.

To address this problem, [3] proposed the masked language model called BERT. Instead of predicting the word after a given sequence, it learns in the process of first informing the model of the entire sequence and then predicting which word corresponds to the (masked) blank. However, there is a disadvantage that it cannot be easily used for summary generation.

To improve abstractive summarization, [9] recently proposed BART, a pre-trained sequence-to-sequence model that combines both BERT as the encoder and GPT as the decoder. BERT and GPT are transformers for bidirectional masked and auto-regressive language models, respectively. In particular, BART is a denoising autoencoder model. Specifically, it is trained by first corrupting text with one of arbitrary noising functions such as token masking, token

---

**FIGURE 2. Overview of BART [9].**
deletion, text infilling, sentence permutation, and document rotation, and then learning the model to reconstruct the original text.

As shown in Figure 2, the corrupted text created by one of the noising functions is first encoded with a bidirectional encoder and then a summary is generated with an autoregressive decoder. For pre-training, the input of the encoder is the corrupted text, while the input of the decoder is the original text. For fine-tuning, the input of the encoder is the original text, while the input of the decoder is the reference summary.

In BART, the following noising functions are mainly used.

- **Token Masking (TM):** Some tokens are sampled at random and replaced with [MASK] tokens.
- **Token Deletion (TD):** Randomly sampled tokens are deleted from the original text.
- **Token Infilling (TI):** BART does span masking, not token masking in BERT. For example, a text span of three tokens is replaced with a single [MASK] token.
- **Sentence Permutation (SP):** A original text is decomposed into multiple sentences, and then the sentences are shuffled in random order.
- **Document Rotation (DR):** After the original text is shuffled, a token chosen at random is appended to the beginning of the text. This trains the model to identify the beginning of the text.

According to Lewis et al.’s experimental result, both TI and SP showed the best performance for text summarization. Thus, we used them as the noising function in our experiments.

In addition, Lewis et al. reported that BART outperformed main state-of-the-art summarization models including Lead-3, PTGEN [6], PTGEN+COV [6], UniLM [7], BERT-SUMABS [8], and BERTSUMEXTABS [8]. Therefore, we consider BART as the baseline model in this article.

B. PROPOSED METHOD

Our proposed model for abstractive summarization is based on BART as described in Section III-A. We extend the existing BART model by adding two components: (i) discriminator $\Delta$ and (ii) cluster generator $\tau$. The pre-training step is the same as the existing BART model. However, the fine-tuning step is quite different. Given an original text as input, the discriminator first splits the original text into a set of sentences and then classifies whether each sentence is salient or not. The cluster generator takes as input context vectors from the encoder in BART and forms two clusters: One is the group of the context vectors corresponding to salient sentences, while the other is the group of the context vectors corresponding to non-salient sentences. The decoder in BART learns these cluster patterns for summary generation and predicts words auto-regressively.

Figure 3 illustrates the overview of the proposed model for abstractive summarization. The original text document $d$ consists of five sentences $d = \{s_1, s_2, s_3, s_4, s_5\}$. $\Delta$ takes $d$ as input and classifies whether each sentence is salient or not. For example, if $\Delta$ classifies $s_1$ and $s_3$ to be salient, while classifying $s_2$, $s_4$, and $s_5$ to be non-salient, we can have two clusters: $\{s_1, s_3\}$ and $\{s_2, s_4, s_5\}$. The output of the encoder in BART is a set of the context vectors corresponding to the sentences in $d$. When $\vec{v}_i$ is a context vector corresponding to sentence $s_i$, we can see two clusters of the context vectors: $\{\vec{v}_1, \vec{v}_3\}$ and $\{\vec{v}_2, \vec{v}_4, \vec{v}_5\}$.

Based on this cluster information, the cluster generator converts existing context vectors $\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4$, and $\vec{v}_5$ to new context vectors $\vec{v}_1', \vec{v}_2', \vec{v}_3', \vec{v}_4'$, and $\vec{v}_5'$, satisfying the conditions below:

- $\vec{v}_1' \approx \vec{v}_3'$
- $\vec{v}_2' \approx \vec{v}_4' \approx \vec{v}_5'$
- $\{\vec{v}_1', \vec{v}_3'\}$ is far apart from $\{\vec{v}_2', \vec{v}_4', \vec{v}_5'\}$.

C. DISCRIMINATOR $\Delta$

For $\Delta$, we use the pre-trained BERT with transformer layers, borrowing similar idea from [8]. Figure 4 shows the neural architecture of $\Delta$ by adding transformer encoders for classification as two layers on top of the existing BERT. The $\Delta$ divides the input text into sentences and adds a [CLS] token at the beginning of each sentence. Among the token vectors through BERT, only [CLS] token vectors are selected and transferred to the transformer encoders. Finally, the [CLS] token vectors, which contain the representation of each sentence, are output as values of 0 to 1. The higher the output value, the more important the sentence. The sentences with high value are considered to be salient.

$$C_p, C_n \leftarrow \Delta(d, \lambda)$$ (1)

In Eq. 1, $C_p$ is the set of salient sentences, while $C_n$ is the set of non-salient sentences. As the input of $\Delta$, $\lambda$ is a hyper-parameter controlling the number of salient sentences in $d$. This $\lambda$ value can be different across various data sets. In our experiments, we find the optimal $\lambda$ by investigating ROUGE results in different $\lambda$ values.

D. CLUSTER GENERATOR $\tau$

The goal of this layer is to transform a context vector $\vec{v}_i$ from the encoder neural model in BART to a new vector representation $\vec{v}_i'$ including clustering information. Please take a look at the following Eq. 2.

$$\vec{v}_i' \leftarrow \tau(\vec{v}_i),$$ (2)

where $\tau()$ is the function about the cluster generator.

Figure 5(a) depicts four contextualized vectors $\vec{v}_1, \vec{v}_2, \vec{v}_3$, and $\vec{v}_4$ in the latent space.\footnote{We simplify the number of vector dimensions as two for intuitively understanding. In the coordinates, $d_1$ and $d_2$ are the first dimension and the second one, respectively.} Supposing that a given original text $d$ consists of four sentences $s_1, s_2, s_3$, and $s_4$, e.g., $d = \{s_1, s_2, s_3, s_4\}$, $\vec{v}_i$ is the context vector of $s_i$ and $|\vec{v}_i|$ is the length of $\vec{v}_i$. In the figure, the red vectors (e.g., $\vec{v}_1$ and $\vec{v}_2$) are labeled as salient sentences from $\Delta$, while the blue ones (e.g., $\vec{v}_2$ and $\vec{v}_4$) are non-salient ones.
FIGURE 3. Overview of the proposed model (Salient sentences in red and non-salient ones in blue).

FIGURE 4. Discriminator Δ.

1) CLUSTER GENERATION
To cluster salient and non-salient sentences, the representation of the context vectors corresponding to the salient sentences is recalculated by the following equation.

$$\vec{v}_i^{(l)} = \begin{cases} \tau_1(w \cdot \vec{v}_i) & \text{if } \vec{v}_i \in C_p \\ \tau_1(\vec{v}_i) & \text{if } \vec{v}_i \in C_n \end{cases}$$

where $w$ is a certain weight value that is a hyper-parameter and we used the optimal $w$ through a series of experiments with different $w$ values. As shown in Figure 5(b), when $w = -1$, $\vec{v}_i^{(l)}$ is located in the opposite direction of $\vec{v}_i$ in the latent space. In the figure, context vectors corresponding to non-salient sentences exist in the first quadrant of latent space coordinates, and ones corresponding to salient sentences move to the third quadrant. As a result, two clusters can be formed in the latent space. Now vectors $\vec{v}_1^{(l)}$ and $\vec{v}_3^{(l)}$ are changed to new vector representations containing clustering information.

2) CLUSTER NORMALIZATION
In addition, $C'_p = \{\vec{v}_1^{(l)}, \vec{v}_3^{(l)}\}$ can be further normalized into $C''_p = \{\vec{v}_1^{(l+1)}, \vec{v}_3^{(l+1)}\}$ using Eq. 3 below.

$$\vec{v}_i^{(l+1)} = \tau_2\left(\frac{\vec{v}_i^{(l)}}{\sigma} - \vec{v}_p^\mu\right), \quad (3)$$

where $\vec{v}_p^\mu$ is the mean vector, shown as the black vector in Figure 5(c), and $\sigma$ is the standard deviation in $C'_p$. $C'_n = \{\vec{v}_2^{(l)}, \vec{v}_4^{(l)}\}$ is also normalized into $C''_n = \{\vec{v}_2^{(l+1)}, \vec{v}_4^{(l+1)}\}$. We expect that transformation of the vector representation through Eq. 3 will enable the decoder model in BART to learn cluster patterns more robustly.

3) CLUSTER SHRINKING
Our another approach is to forcibly shrink the size of clusters ($C'_p$ and $C'_n$) that have already been created by Eq. 3. We first obtain the mean vectors $\vec{v}_p^\mu$ in $C'_p$ and $\vec{v}_n^\mu$ in $C'_n$, respectively. Next, for each cluster (e.g., $C'_p$), we compute Euclidean distances between $\vec{v}_p^\mu$ and $\vec{v}_i^{(l+1)} \in C'_p$. For instance, after the distance $dist(\vec{v}_p^\mu, \vec{v}_3^{(l+1)})$ is computed, it is indicated by the dotted line in Figure 5(d). Finally, we divide the distance by the ratio $m$ to $n$. If $m = n = 1$, we move $\vec{v}_3^{(l+1)}$ to the center.
\[ \vec{v}'_i = \frac{\vec{v}_i - \vec{v}_p}{\sigma} w|\vec{v}_i| \]

\[ \vec{v}'_{i+1} = \frac{\vec{v}'_i - \vec{v}'_{i+1}}{\sigma} w|\vec{v}'_{i+1}| \]

where \( m \) and \( n \) are hyper-parameters and we discovered the optimal ratio value through a series of experiments with different ratios of \( m \) to \( n \). This approach can make it easier for the decoder model in BART to learn distinct cluster patterns because it reduces the variance between vectors in a cluster by making the vectors in the cluster close together at a constant rate.

IV. EXPERIMENTAL SET-UP

For the experiment, we first implemented the discriminator and the cluster generator using Python and then combined them into existing BART base model. To execute the BART model, we set 0.1 to dropout rate, 1 to batch size, 3e-5 to learning rate, 2 to gradient accumulation, and 0.1 to gradient clipping. As the hyper-parameters for summary generation, we set 6 to the number of beams, 62 to max length, 3 to max repeat n-gram, and TRUE to early stopping. We used 1,024 tokens as the maximum input size of BART.

To evaluate the proposed model for abstractive summarization, we used two distinct data sets: One is CNN/DailyMail [2] and the other is XSUM [35], which are...
main benchmark data sets in the text summarization field. It is known that the characteristics of the two data sets are slightly different. Reference summaries look like extractive style summaries on the CNN/DailyMail data set. On the other hand, in the XSUM data set, the reference summaries are extremely abstractive summaries, consisting of one or two sentences. On the other hand, each reference summary of the XSUM data set were paraphrased of salient sentences taken directly from the original text.

All models were in standalone executed in a high-performance workstation server with Intel Xeon Scalable Silver 4414 2.20GHz CPU with 40 cores, 24GB RAM, 1TB SSD, and GEFORCE RTX 3080 Ti 11GB BLOWER with 4,352 CUDA cores, 12GB RAM, and 7 GBPS memory clock.

V. EXPERIMENTAL RESULTS
A. COMPARISON OF PROPOSED MODEL TO BART AND PEGASUS

We first tested PEGASUS, BART, and the proposed model with XSUM and CNN/DailyMail data sets, respectively, and then measured ROUGE-1/2/L and BERTScore values. Tables 2 and 3 summarize the results in order to compare the performance of the proposed model with the existing BART and PEGASUS models. After we ran the proposed model with different hyper-parameters such as $\lambda$, $w$, and the ratio of $m$ to $n$ and chose them showing the best ROUGE and BERTScore, we obtained $\lambda = 0.5$, $w = -1.0$, $m = 3$, and $n = 1$ in XSUM data set and $\lambda = 0.6$, $w = -1.0$, $m = 6$, and $n = 1$ in CNN/DailyMail data set.

Compared with BART, the proposed model largely improves ROUGE-1/2/L/avg by 4%/5%/4%/4% and BERTScore by 0.4% in XSUM and improves ROUGE-1/2/L/avg by 2%/2%/2%/2% and BERTScore by 0.1% in CNN/DailyMail. Compared with PEGASUS, the proposed model largely improves ROUGE-1/L/avg by 8%/6%/4% and BERTScore by 0.8% in XSUM and improves ROUGE-1/L/avg by 8%/30%/10% and BERTScore by 0.7% in CNN/DailyMail. The reason why the proposed model shows higher performance than the existing BART and PEGASUS models is that the decoder generates an abstractive summary after learning cluster patterns made by the proposed cluster generator, which creates two clusters for salient and non-salient context vectors and increases the margin between the two clusters. For fine-tuning, the decoder is less affected by noise information such as non-salient context vectors, attending more over the cluster including salient context vectors. As a result, it can generate an abstractive summary including the main content of the original text.

Interestingly, whatever the model is, all the ROUGE scores in the CNN/DailyMail data set are slightly higher than those in the XSUM data set. For example, the ROUGE-avg score of the proposed model is 27.72 in XSUM, whereas it is 33.62 in CNN/DailyMail. In contrast, all the BERTScore values in the CNN/DailyMail data set are slightly lower than those in the XSUM data set. For instance, the BERTScore value of the proposed model is 90.39 in XSUM, whereas it is 88.30 in CNN/DailyMail. This difference is due to the characteristics of the two data sets. Actually, most of the reference summaries of the XSUM data set were paraphrased from the original text as an extreme abstractive summary consisting of one or two sentences. On the other hand, each reference summary of the CNN/DailyMail data set is the set of salient sentences taken directly from the original text.

Although the proposed model outperforms PEGASUS in overall performance, it is not better than PEGASUS in ROUGE-2. For example, as shown in Table 2, PEGASUS shows about 11% higher score than the proposed model. This is because there is a lot of overlap of bigram tokens between the generated summary and the reference summary. The strength of the proposed model based on BART lies in better understanding the overall context and maintaining consistency. In contrast, PEGASUS learns by masking important sentences and predicting them containing key information. For instance, here is an example.

- **Original text**: A major breakthrough in renewable energy technology was announced at the Global Tech Innovators Conference. Industry leaders presented a new solar panel that is twice as efficient as existing models and considerably cheaper to produce. Experts predict this innovation could revolutionize energy consumption worldwide.

| Text document | Reference summary |
|----------------|-------------------|
| Army explosive experts were called out to deal with a suspect package at the offices on the Newtownards Road on Friday night. Roads were sealed off and traffic diverted as a controlled explosion was carried out. The premises, used by East Belfast MP Naomi Long, have been targeted a number of times. Most recently, petrol bomb attacks were carried out on the offices on consecutive nights in April and May. The attacks began following a Belfast City Council vote in December 2012 restricting the flying of the union flag at the City Hall. Condemning the latest hoax, Alliance MLA Chris Lyttle said: “It is a serious incident for the local area, it causes serious disruption, it puts people’s lives at risk, it can prevent emergency services reaching the area. Ultimately, we need people with information to share that with the police in order for them to do their job and bring these people to justice.” | A suspicious package left outside an Alliance Party office in east Belfast has been declared a hoax. |
### B. RESULTS OF DIFFERENT WEIGHT VALUES

As already discussed in Section III-D1, by scaling a context vector by a certain weight value \( w \), context vectors corresponding to salient and non-salient sentences are clustered in the latent space. If \( w \) is between 0 and 1, the length of new vector \( \vec{v}_i^{(l)} \) is reduced, compared to the given context vector \( \vec{v}_i \), but the direction is the same. Furthermore, if \( w \) is greater than 1, the length of \( \vec{v}_i^{(l)} \) increases but the direction is also the same. On the other hand, if \( w \) is between 0 and -1, the length of \( \vec{v}_i^{(l)} \) is small, indicating that it is in the opposite direction, compared to \( \vec{v}_i \). If \( w = -1 \), new vector \( \vec{v}_i^{(l)} \) has the same length but the opposite direction.

If the cluster of salient sentence vectors is far from the cluster of non-salient ones in the latent space, the decoder in BART can focus mainly on learning the salient sentence vectors out of non-salient ones. For this reason, selecting an appropriate weight value \( w \) is important in the proposed model. In this experiment, we performed the proposed model with different weight values \( w = \{-2.0, -1.5, -1.0, 1.5, 2.0\} \) in the XSUM data set and measured its ROUGE-1/2/L and BERTScore values as summarized in Table 4.

When \( w = -1.0 \), the proposed model shows the best results in both ROUGE-1/2/L and BERTScore. These results are reasonable because of the relative position between the two clusters in the latent space. For any vector \( \vec{v}_i \), the proposed model can generate the best summary when new vector \( \vec{v}_i^{(l)} \) has the same length as \( \vec{v}_i \) and is located in the opposite direction in the latent space. In other words, when \( w = -1 \), two clusters are close to each other, so they may not be well distinguished in the latent space. In contrast, the larger \( w \), the farther the distance between two clusters becomes, so their relevance decreases and performance deteriorates. Note that

### TABLE 4. ROUGE-1/2/L and BERTScore values of the proposed model according to different weight values.

| \( w \) | ROUGE-1 | ROUGE-2 | ROUGE-L | BERTScore |
|--------|---------|---------|---------|-----------|
| -2.0   | 37.34   | 14.90   | 29.90   | 90.24     |
| -1.5   | 37.46   | 14.87   | 29.96   | 90.26     |
| -1.0   | 37.54   | 14.96   | 30.04   | 92.29     |
| 1.5    | 37.06   | 14.71   | 29.40   | 90.16     |
| 2.0    | 37.04   | 14.61   | 29.28   | 90.10     |

### TABLE 5. ROUGE-1/2/L and BERTScore values of the proposed model according to different \( \lambda \) values.

| \( \lambda \) | ROUGE-1 | ROUGE-2 | ROUGE-L | BERTScore |
|---------------|---------|---------|---------|-----------|
| 0.2           | 37.84   | 15.26   | 30.20   | 90.33     |
| 0.3           | 37.68   | 14.99   | 30.07   | 90.36     |
| 0.4           | 37.74   | 14.98   | 30.04   | 90.30     |
| 0.5           | 37.81   | 15.07   | 30.27   | 90.39     |
| 0.6           | 37.55   | 15.10   | 30.05   | 90.27     |
| 0.7           | 37.40   | 14.91   | 29.96   | 90.30     |

- **Reference summary**: At the Global Tech Conference, a new, more efficient, and cheaper solar panel was introduced that may change energy consumption globally.
- **Summary generated by PEGASUS**: The Tech Innovators Conference highlighted a solar panel breakthrough, promising to enhance energy efficiency and reduce costs.
- **Summary generated by Proposed Model**: At this year’s Global Tech Conference, scientists talked about global warming’s impact on ice caps and proposed new emission reduction strategies.

In the reference summary, key information is “more efficient, and cheaper solar panel”. The summary generated by PEGASUS contains key information about “solar panel breakthrough”, “energy efficiency”, and “reduce cost”, which connect “more efficient, and cheaper solar panel” well. However, key information like “global warming’s impact” in the summary generated by the proposed model disappears in the reference summary. This generated summary has partially similar meanings, but does not contain key information like the summary of PEGASUS.
TABLE 7. A case study of summaries generated by BART and the proposed model.

| Text document                                                                 |
|--------------------------------------------------------------------------------|
| A spokesman for Palm Beach Gardens police in Florida confirmed to the BBC they  |
| were investigating a fatal crash involving the Grand Slam champion. A man      |
| was taken to hospital after the accident on 9 June and died two weeks later   |
| from his injuries, he said. According to TMZ, which broke the story, police    |
| believe the seven-time Grand Slam champion was at fault. But a lawyer from     |
| Williams said it was an “unfortunate accident”. The man who died, Jerome      |
| Barson, was travelling with his wife who was driving their vehicle through an  |
| intersection when the accident happened. Williams’s car suddenly darted into  |
| their path and was unable to clear the junction in time due to traffic jams,   |
| according to witness statements in a police report obtained by US media. Mrs   |
| Barson was also taken to hospital but survived. “Williams is at fault for     |
| violating the right of way of the other driver.” The report said, adding that  |
| there were no other factors like drugs, alcohol or mobile phone distractions.  |
| The 37-year-old tennis star reportedly told police she did not see the couple’s  |
| car and she was driving slowly. Police spokesman Major Paul Rogers said police |
| were investigating whether the incident was connected to Mr Barson’s death.     |
| William’s lawyer Malcolm Cunningham told CNN in a statement: “Ms Williams      |
| entered the intersection on a green light. The police report estimates that Ms |
| Williams was travelling at 5mph when Mrs Barson crashed into her. “Authorities |
| did not issue Ms Williams with any citations or traffic violations. This is an |
and non-salient vectors. Through this clustering shrinking method, the decoder in BART are likely to learn cluster patterns easily.

Table 6 shows the results of the proposed model according to different ratios of $m$ to $n$ values, where $w$ and $\lambda$ are set to $-1.0$ and $0.5$. In the XSUM data set, both the ROUGE-1/2/L and BERTScore values are the highest when $m = 3$ and $n = 1$. On the other hand, in the case of the CNN/DailyMail data set, best ROUGE-1/2/L and BERTScore results are shown when the ratio of $m = 6$ to $n = 1$. For instance, the ROUGE-1/2/L and BERTScore values of the proposed model are 41.99, 20.01, 38.85, and 88.30, respectively. This indicates that the ratio of $m$ to $n$ depends on the characteristics of the given summarization data set for cluster shrinking.

E. DISCUSSION OF GENERATED SUMMARIES

Table 7 shows two abstractive summaries generated by both BART and the proposed model. The actual original text has a long text span, but due to space limitations, we omitted a significant amount of text in the table. As the reference summary, the main point summarized by a human evaluator is that US tennis star Venus Williams was involved in a car accident with the death of a man. Both summaries generated by BART and the proposed model focus on a car accident with a famous tennis player in US. The BART and proposed models all summarized the original text well as a human-written abstract usually does. In addition, the generated summaries are natural, fluent, and no errors in grammar.

However, the BART model summarized incorrectly, while the proposed model did correctly. The US tennis star is not Serena Williams but Venus Williams in the car accident. As we can see from these results, learning distinct cluster patterns in the latent space prevents the sequence-to-sequence model for abstractive summarization from inaccurately reproducing factual details. This is because the summary is generated by referring to only salient sentences that are well representative of the given text document. In addition, these results might give us an indirect evidence in that the $\Delta$ and $\tau$ of our proposed method work well.

VI. CONCLUSION

To improve abstractive summarization, we propose a new pre-trained sequence-to-sequence model containing a discriminator and a cluster generator between the encoder and the decoder. During fine-tuning, the discriminator extracts salient sentences from a given text document. The cluster generator generates two clusters of salient and non-salient context vectors from the encoder, and normalizes and shrinks the clusters to better distinguish the two clusters. Our experimental results show that the proposed method outperforms the existing BART and PEGASUS models in two summarization benchmark data sets.

For the future research direction, we will further refine our proposed model in order to tackle the fact inconsistency (hallucination) problem between input document and generated summary and to generate abstractive summaries for long documents such as scientific papers, reports, and books.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their feedback.

REFERENCES

[1] A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 379–389.
[2] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, “Abstractive text summarization using sequence-to-sequence RNNs and beyond,” in Proc. 20th SIGMUL Conf. Comput. Natural Lang. Learn., 2016, pp. 1–12.
[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol., 2019, pp. 4171–4186.
[4] J. Demsg, Tech. Rep.2018.
[5] S.-G. Jo and B.-W. On, “Weighted context vector-based BART model for improving abstractive summarization,” J. Korean Inst. Inf. Technol., vol. 19, no. 10, pp. 19–27, Oct. 2021.
[6] A. See, P.-J. Liu, and C. D. Manning, “Get to the point: Summarization with pointer-generator networks,” in Proc. 55th Ann. Meeting Assoc. Comput. Linguistics, 2017, pp. 1073–1083.
[7] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, “Unified language model pre-training for natural language understanding and generation,” 2019, arXiv:1905.03197.
[8] L. Gao and M. Lapata, “Text summarization with pretrained encoders,” 2019, arXiv:1908.03845.
[9] S.-G. Jo et al.: Learning Cluster Patterns for Abstractive Summarization
[10] A. Sahu and S. G. Sanjeevi, “Better fine-tuning with extracted important sentences for abstractive summarization,” in Proc. Int. Conf. Commun., Control Inf. Sci. (ICCISc), Jun. 2021, pp. 11328–11339.
[11] Y.-C. Chen and M. Bansal, “Fast abstractive summarization with reinforce-selected sentence rewriting,” in Proc. 56th Ann. Meeting Assoc. Comput. Linguistics, 2018, pp. 675–686.
[12] R. Paulus, C. Xiong, and R. Socher, “A deep reinforced model for abstractive summarization,” in Proc. Int. Conf. Learn. Repres., 2018, pp. 1–13.
[13] Z. Wang, Z. Duan, H. Zhang, C. Wang, L. Tian, B. Chen, and M. Zhou, “Friendly topic assistant for transformer based abstractive summarization,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020, pp. 485–497.
[14] P. Cui and L. Hu, “Topic-guided abstractive multi-document summarization,” in Proc. Findings Assoc. Comput. Linguistics, 2021, pp. 1463–1472.
[15] T. Yu, W. Dai, Z. Liu, and P. Fung, “Vision guided generative pre-trained language models for multimodal abstractive summarization,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2021, pp. 3995–4007.
[16] S. Cao and L. Wang, “Attention head masking for inference time content selection in abstractive summarization,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol., 2021, pp. 5008–5016.
[17] L. Xiao, L. Wang, H. He, and Y. Jin, “Modeling content importance for summarization with pre-trained language models,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020, pp. 3606–3611.
[18] N. I. Nikolov and H. R. Hahnloser, “Abstractive document summarization without parallel data,” in Proc. 12th Conf. Lang. Resour. Eval., 2020, pp. 6638–6644.
[19] C. Jumel, A. Louis, and J. C. K. Cheung, “TESA: A task in entity semantic aggregation for abstractive summarization,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020, pp. 8031–8050.
