GAN based Korean summarizer using semi abstractive method

Hoon-Suk Lee
Department of Computer Engineering,
Dankook university
dolmani38@dankook.ac.kr
hoonsl@asianaidt.co.kr

Soon-Hong An
Department of Computer Engineering
Dankook university
kingmir@dankook.ac.kr
ansh@asianaidt.com

Seung-Hoon Kim
Department of Computer Engineering
Dankook university
edina@dankook.ac.kr

***Abstract*— Recently, the NLP field is making a leap forward with various transformer models that is come out through encoder, decoder and attention. Document summarization is also possible for abstract summarization by transformer. However, in the case of Korean language, the development of abstract summary models is slow because there is no high-quality mass amount of summary data set. An extractive summary method as an unsupervised learning can be used, but it has the disadvantage that an isthmus summary is made rather than inclusion of the overall content. In order to overcome these obvious flaw, In this paper, we proposes a semi-abstractive method (SAM) that derives words necessary for summary from the original text and generates a new abstractive summary by GAN without a large amount of summary data set. Experiments show that the proposed method has the advantage of maintaining the similarity, solving the isthmus problem and grammatical consistency of the entire document through GAN learning.**

Keywords—NLP, Transformer, GAN, Summarization

I. INTRODUCTION

When you open a news site, not all news articles are read from start to finish. In general, go through a short news summary and then read the details if you are interested. Short and informative summaries of news are available everywhere, such as in recent magazines, news aggregator apps, and research sites. The method of extracting these summaries from the original large text without losing important information is called text summaries. The summary is grammatically easy to read and it is essential to describe what is important. In fact, Google News and many other news aggregator apps utilize text summarization algorithms.

Text summary methods can be grouped into two main categories: extractive methods and abstractive methods.

A. *Extractive text summary*

It identifying important sections of the text and generating them verbatim producing a subset of the sentences from the original text. This method is widely used in that it contains the exact sentence of the original text. However, it has an isthmus problem in which only part of the document is summarized.

B. *Abstractive text summary*

It is a more advanced method, and a lot of research is currently underway. The approach is to identify important sections, interpret context, and reproduce in new sentences. As a result, a new summary sentence is created, not the sentence extracted from the original text. However, the recent Transformer method requires a large amount of summary data sets.

In this paper, as an abstractive text summary method, the original text is summarized by deriving and combining tokens, not sentences, from the original text, generating new sentences including the proper grammatical flow and contents of the original text.

II. RELATED WORK

First, related studies on the extraction method are as follows.

Luhn (Hans Peter Luhn, 1960) is a method of extracting the highest scored sentences by scoring the importance of sentences based on TF-IDF (Term Frequency-Inverse Document Frequency). By developing this concept, the TextRank (Rada Mihalcea et al., 2004) method and the LexRank (G¨une¸s Erkan, 2004) extraction algorithm were developed. The concept of machine learning began to be introduced in LSA (Susan T. Dumais, 2005) and KL-Sum (Aria Haghighi, 2009), and recently, studies are actively applying BERT and transformer models to extraction techniques. Ming Zhong proposed the MATCHSUN framework (Ming Zhong et al., 2020), using CNN/DailyMail as a dataset, and recording a ROUGE1 score of 44.41. The BERTSUM (Yang Liu et al., 2019) method can be applied to both the Extractive and Abstractive methods, and the ROUGE1 score of 43.25 was recorded as the extractive method. The results of these recent studies are that abstractives are not unconditionally superior to extractives. In other words, it will be possible to use an appropriate method according to the purpose and conditions.

Abstract summarization is a new state-of-the-art method of generating new sentences that best represent the entire text. This produces a content closer to human summary than the extraction method of selecting sentences from the original text for summary purposes.

For abstract summaries, in fact, the context of the original text must be grasped. An important step in understanding the context in NLP's research is the sequence-to-sequence (seq2seq) and Attention model, which was released in 2014. (Sutskever et al., 2014; Cho et al., 2014). Numerous concepts have been studied based on this model. After that, the Transformer model was introduced by Google in 2017 and received great attention from NLP academia. This is because we were able to improve machine translation performance to

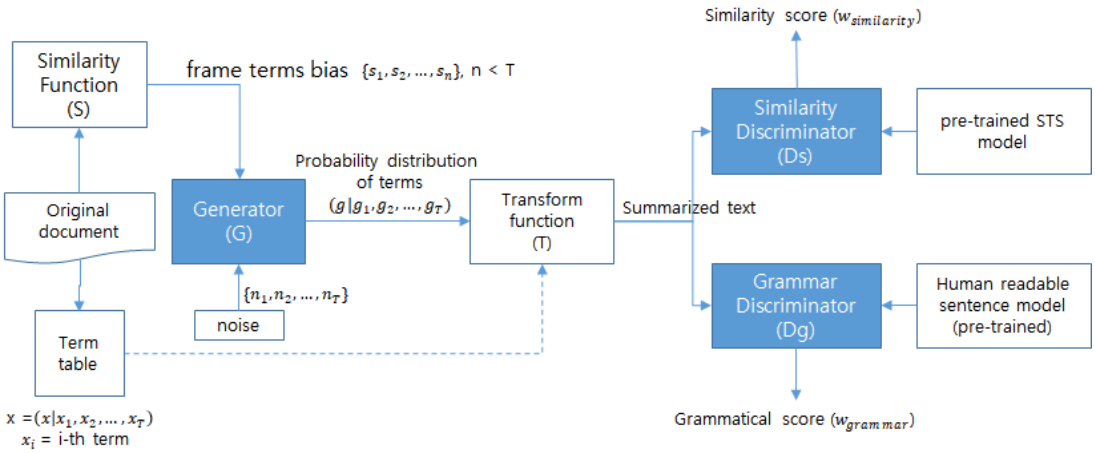

Figure 1 Overall architecture of GAN based SAM for Korean

the next level, following Seq2seq and Attention, by proposing a new model, breaking away from the studies that were dominated by the existing CNN and RNN.

Abstract summarization is treated as a kind of machine translation that turns long sentences into short sentences. Therefore, in general, abstract summary methods are applied in various Transformer models such as T5(Colin Raffel et al.,2020), BART Transformer (Mike Lewis et al., 2019), GPT-2 Transformers (Alec Radford et al., 2018), XLM Transformers (Guillaume Lample et al., 2019).

Looking at the original text as the 'original' and the summary as the concept of 'imitation' of the original text, the GAN learning algorithm can also be said to be a methodology that forms the main framework of the abstract summary.

There have been studies of applying the GAN algorithm to make the output of the sentence generator human-readable. The biggest problem in applying GAN to sentence generation is the discrete nature of natural language. To generate a word sequence, the generator usually has parts such as non-differences, such as argmax or other sample functions that cause failures against the original GAN algorithm. In (Gulrajani et al., 2017), the author provides a generator output layer directly to the discriminator instead of providing individual word sequences. This method is effective because it uses the Earth Mover (EM) distance in GAN, as suggested by (Arjovsky et al., 2017), which allows you to evaluate the distance between the discrete and continuous distributions. SeqGAN (Yu et al., 2017) deals with sequence generation problems with reinforcement learning. This approach is referred to here as hostile REINFORCE. However, since the discriminator only measures the quality of the entire sequence, the rewards are extremely rare and the rewards assigned to all stages of creation are all the same. MC search (Yu et al., 2017) was proposed to evaluate the approximate reward at each time step, but this method has a high time complexity. Following this idea (Li et al., 2017) had proposed a partial assessment approach to assess the expected reward at each time step.

As discussed above, it was conceived that unsupervised learning abstract summarization could be realized through merging of BERT and GAN techniques. In the next section, we describe a summary method proposed in this paper that implements a discriminator using a BERT and trains a generator using a GAN technique.

III. Proposed Method

The method proposed in this paper can be called a semi abstractive method that combines the extractive and the abstractive method. Divide the original document by token and reassemble it to create several sentences that summarize the entire contents.

For this purpose, the Generator receives noise as inputs as much as the number of tokens of the original document and has outputs as the same number with inputs. And then, the output corresponds to the probability value of that tokens used for summarizing original document. Therefore, a summary sentence is generated by selecting a token corresponding to a specific probability value or a probability value of the highest ranking corresponding to the number of given tokens. Frame tokens are extracted from the original document by similarity function and these are input as bias to the generator. Through this, the Generator first allocates a high probability values to the frame tokens, and generate sentences randomly somewhat similar to original document and then identifies whether the sentences are grammatically human-readable sentences by the Korean grammar discriminator. Also, the similarity, which distinguishes the similarity of the original document, is adjusted so that the randomly generated sentences by the discriminator do not differ from the original document. The overall architecture is shown in the figure1.

A. Frame tokens extraction using similarity function

Split the document into token units (divided by space units). When each token is called x_i, the document can be expressed as follows.

$$x = \{x_1, x_2, \dots, x_T\} \quad (1)$$

The Similarity Probability Distribution (SPD) of the entire document for each word can be expressed as follows.

$$SPD_{for\ one\ term} = \{P_s(x_1), P_s(x_2), \dots, P_s(x_T)\} \quad (2)$$

This is regarded as a continuous signal, and tokens corresponding to the peaks of the signal are extracted, and these are composed of frame tokens corresponding to the summary of the entire document.

However, in the entire story of document, for example, a word that refers to the protagonist has a dominant effect on the overall similarity, and the peaks of the signal can consist only of the corresponding tokens. Eventually, the desired frame cannot be obtained. As a way to overcome this, the SPD of a story line composed of multiple tokens is calculated. If it is two tokens, the similarity probability is as follows.

$$P_s(x_i, x_{i+1}) = S(x, (x_i, x_{i+1})) \tag{3}$$

Here, (x_i, x_{i+1}) is a kind of Partial story that acts as a filter and convolutions the entire document. If you use N tokens as a filter, The SPD becomes like that

$$SPD_{for\ N\ term} = \{P_s(x_1, x_2, \dots, x_N), P_s(x_2, x_3, \dots, x_{N+1}), \dots, P_s(x_{T-N}, x_{T-N+1}, \dots, x_T)\} \tag{4}$$

If frame tokens = s are composed as follows using m filters

$$s = peak(\sum_{i=1}^{m} SPD_i) \tag{5}$$

$$s = \{s_1, s_2, \dots, s_n\}, n < T \tag{6}$$

However, the problem here is that even in $SPD_{1\sim m}$, the peaks are concentrated in the dominant token like the main character, and the frame corresponding to the story line may not be extracted. To overcome this, before extracting the peak, the value of SPD_1 is subtracted to avoid the peak of the dominant token, and the frame corresponding to the story line can be extracted.

$$s = peak(\sum_{i=2}^{3} SPD_i) - peak(SPD_1) \tag{7}$$

B. Similarity function, S (Sentence-BERT)

Sentence-BERT(Nils Reimers et al.,2019), a modification of the pre-trained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity.

In this paper, the 'xlm-r-large-en-ko-nli-ststb' was used for pre-trained model in Korean , and the STS (Semantic Textual Similarity) Benchmark score of this model was 84.05%.

Similarity function was implemented using sentence-transformer package, a python library implementing SBERT. Similar to BERT-based models, this library has an input limit. The input accepts a maximum of 128 tokens and returns a 1024-dimensional embedding vector. That is, it is not possible to embedding the entire document at once, and to solve this problem, the document is divided by sentence and embedding in a matrix of (N, 1024) for N sentences. After matrixing the sentences for measuring similarity (n,1024), the cosine distance is calculated pair-wisely to obtain the (n,N) matrix. After that, the average value of the minimum value for each row was taken, and the similarity for the entire document was calculated.

C. Korean Grammar Discriminator, D

In order to classify the consistency of Korean grammar, we construct BERT-based sentence classification. About 30,000 Korean single sentences were fine-tuned based on the Korean pre-trained model 'monologg/kobert'. For normal sentences, label = 1, grammatically abnormal sentences were labeled as label = 0, and abnormal sentences were made by simply shuffling the normal sentences. As a result of measuring the performance with epoch 4 times and 3,000 validation sets, an F1 score of 0.99 was obtained.

D. Korean Text Generator, G

Text Generator used general DNN for simplification. Random noise and frame tokens corresponding to the total number of tokens in the original document are input as a bias. That is, there are two input terms and the output is equal to the number of tokens in the original document as it is the same as the input because we do not know which token will be used for the summary.

Random noise goes through several dense layers, but frame tokens bias is added to the output tensor of the deep network just before the output of the same dimension. As a result, the entire output is biased to the frame tokens, and a sentence summarizing the original document is generated. Also, appropriate tokens between each frame tokens are extracted from the original document by random noise. Then, in the generated probability distribution (Output), the token to be used in the summary is selected according to the following conditions.

$$t'_j = \begin{cases} i & if\ g_i > \alpha \\ None & otherwise \end{cases} \tag{8}$$

Here, you can adjust the length of the generated text by adjusting the α value. A text that can be read grammatically is generated through an appropriate α value. Finally, the selected t'_j vector corresponds to the order selected from x, and text is created by the $x[t']$ operation. Figure 2 below describes the overall Generator structure.

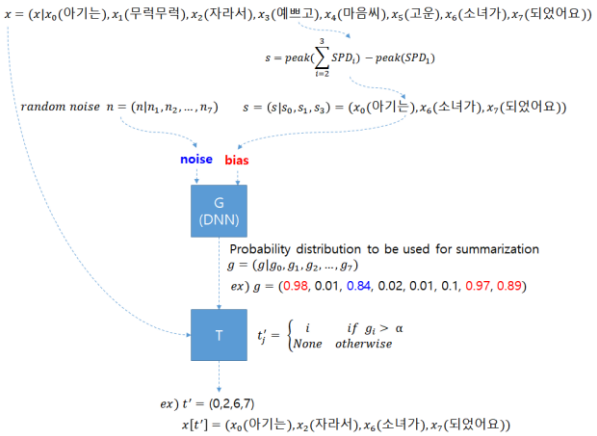

Figure 2 Diagram of text generator

E. WGAN Training

In GAN's paper (Ian J. Goodfellow at el,. 2014) the process that satisfies the following for value function V(G,D) learning is achieved by

$$\min_G \max_D V(D,G) = E_{x\sim P_{data(x)}}[logD(x)] + E_{z\sim P_{data(z)}}[\log(1-D(G(z)))] \quad (9)$$

In other words, it can be seen that it is a process of optimization for cross-entropy $E_{x\sim P_{data(x)}}[logD(x)]$ for D and cross-entropy $E_{z\sim P_{data(z)}}[\log(1-D(G(z)))]$ for G.

The overall objective function covered in this paper can be expressed as follows.

$$\min_G \max_{D_g,D_s} V(D_g,D_s,G) = E_{x\sim P_{data(x)}}[logD_g(x)] + E_{x\sim P_{data(x)}}[logD_s(x)] + E_{z\sim P_{data(z)}}[log(1-D_g(T(G(z))))] + E_{z\sim P_{data(z)}}[log(1-D_s(T(G(z))))] \quad (10)$$

The problem here is that the transform function T is not differentiable. These problems are frequently encountered in the text generation algorithm through GAN. In the paper (Yau-ShianWang at el,. 2018) 'Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks', 'Self-Critic Adversarial REINFORCE' has been suggested. In this technique, a discrete sequence was supplied to the discriminator, so the slope of the discriminator couldn't be directly back propagated to the generator. Here, the policy gradient method was used.

However, this method is possible only with approaching the time series by RNN in generator, and the application of discounted reward d has a very narrow differential width of the loss, requiring a considerable amount of epoch. Therefore, in this paper, we intend to use the modified wasserstein distance (Martin Arjovsky at el,. 2017) without using cross-entropy as a loss.

For the output of G(z) = $(g|g_0,g_1,...,g_T)$, the T function has the following process inside.

$$t_i = \begin{cases} g_i & if\ g_i > \alpha \\ 0 & otherwise \end{cases} \quad (11)$$

The value of α above is the same as the value of Equation (8), and the number of tokens to be acquired is determined by this. The larger the α value, the fewer the number of tokens in the summary result, and the smaller the α value, the more the number of summary tokens.

An example of the output of T(G(z)) is as follows. $(t|t_0,t_1,...,t_T) \sim (t|g_0,0,0,g_3,0,g_5,...,0,g_T)$

If the scalar output of the discriminator is defined as follows, D(T(G(z))) = w then we can use the wasserstein distance as the loss like this

$$G_{loss} = -\frac{w_D}{T}\sum_i^T t_{i\sim G(z)} \quad (12)$$

As a result, t_i is a part of G's output $(g|g_0,...,g_T)$, which can be differentiated, and learning can be achieved through backward. The overall objective function is as follows.

$$\min_G \max_{D_g,D_s} V(D_g,D_s,G) = E_{x\sim P_{data(x)}}[logD_g(x)] + E_{x\sim P_{data(x)}}[logD_s(x)] - \frac{w_{D_g}+w_{D_s}}{T}\sum_i^T t_{i\sim G(z)} \quad (13)$$

Although each discriminator for measuring grammar and similarity was initially applied using a pre-trained model, it must be trained in the process of GAN. However, in the process of learning $E_{x\sim P_{data(x)}}$ cannot be defined. This is because it is not possible to create summarized real data already. So, after substituting some sentences from the original text, there was a tendency to overfitting as part of the original document as learning progressed. In other words, it converges as a result of the extractive method. So, in this paper, Discriminator was not involved in learning process because the Discriminator has already been learned by the pre-trained model. Therefore, the final objective function is simplified as follows.

$$\min_G V(G) = -\frac{w_{D_g}+w_{D_s}}{T}\sum_i^T t_{i\sim G(z)} \quad (14)$$

IV. Experiment

A. Summary results

Korean Cinderella story was used as a sample document to confirm the summary result by this proposed method. The sample document consists of 1,516 characters and 325 tokens.

Original Korean Cinderella story
옛날 어느 집에 귀여운 여자 아기가 태어났어요. 아기는 무럭무럭 자라서, 예쁘고 마음씨 고운 소녀가 되었어요. 그러던 어느날, 소녀의 어머니가 병이들어 그만 세상을 떠나고 말았어요. 소녀의 아버지는 홀로 남은 소녀가 걱정되었어요. 그래서 얼마 후 새어머니를 맞이했어요. 새어머니는 소녀보다 나이가 위인 두 딸을 데리고 왔어요. 그러나 새어머니와 언니들은 성질이 고약한 심술쟁이들이었어요. 새어머니는 소녀가 자기 딸들보다 예쁘고 착한 게 못마땅했어요. 그런데 이번에는 아버지마저 돌아가셨어요. 소녀는 하녀처럼 하루 종일 쓸고, 닦고, 집안일을 도맡아 했어요. 해도 해도 끝이 없는 집안일이 힘들어 지칠때면 난롯가에 앉아서 잠시 쉬곤 했지요. 어느 날, 왕궁에서 무도회가 열렸어요. 신데렐라의 집에도 초대장이 왔어요. ··· (omit middle) ··· 신데렐라가 허둥지둥 왕궁을 빠져나가는데, 유리 구두 한 짝이 벗겨졌어요. 하지만 구두를 주울 틈이 없었어요. 신데렐라를 뒤쫓아오던 왕자님은 층계에서 유리 구두 한 짝을 주웠어요. 왕자님은 유리 구두를 가지고 임금님께 가서 말했어요. 이 유리 구두의 주인과 결혼하겠어요. 그래서 신하들은 유리 구두의 주인을 찾아 온 나라를 돌아다녔어요. 언니들은 발을 오므려도 보고, 구두를 늘려도 보았지만 한눈에 보기에도 유리 구두는 너무 작았어요. 그때, 신데렐라가 조용히 다가와 말했어요. 저도 한번 신어 볼 수 있나요? 신데렐라는 신하게 건넨 유리 구두를 신었어요, 유리 구두는 신데렐라의 발에 꼭 맞았어요. 신하들은 신데렐라를 왕궁으로 데리고 갔어요. 그 뒤 신데렐라는 왕자님과 결혼하여 오래오래 행복하게 살았대요.

- Frame word extraction result

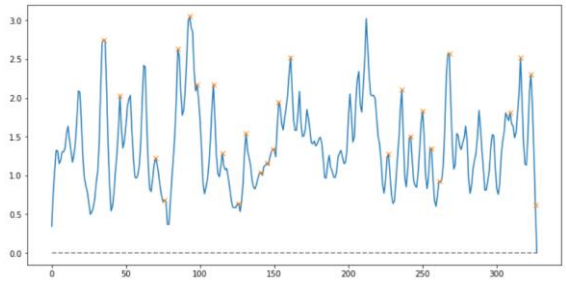

Figure 3 The sample story consisted of 325 tokens, and the result of extracting the peak (marked x) from the similarity distribution for the entire document by all tokens corresponded to the frame tokens.

- WGAN Training result

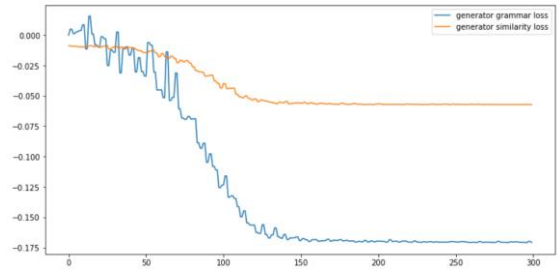

Figure 4 Training was conducted with 300 epoch, the strength of the initial bias for frame tokens was 0.25, learning rate of 5e-5, and Adam optimizer. It was shown that the grammar loss and the similarity loss decrease, respectively, but did not decrease after about 150 epcoh.

- Summary result

Summary of Korean Cinderella story
소녀가 새어머니를 맞이했어요. 위인 언니들은 성질이 했어요. 해도 힘들어 앉아서 잠시 날, 왕궁에서 열렸어요. 언니들을 떠났어요. 훌쩍훌쩍 도마뱀을 오렴. 마법사 황금 생쥐와 생쥐는 마부로 드레스로 마음을 왕자님과 열두 신데렐라는 화들짝 왕궁을 벗겨졌어요. 왕자님은 주웠어요. 유리 임금님께 가서 주인과 결혼하겠어요. 찾아 언니들은 오므려도 보고, 한눈에 다가와 신데렐라는 유리 구두는 발에 맞았어요. 왕궁으로 왕자님과 결혼하여 살았대요.

The compression rate generated as a summary was 16.9% compared to the original document, the similarity of the original document was 60.9%, and the grammar was 98.5%. A new sentence was constructed according to the designed intention, and it contained the overall contents of the original document. However, there were many grammatically unnatural parts, so it was not smooth to read.

B. *Performance comparison with existing summary algorithm*

Since the Korean abstractive summarizer had not existed a comparable object, we compared it with the following three extractive summary method using the Korean BERT Model and verify the advantages of the proposed method.

- SAM+WGAN : proposed method
- BERT+LexRank : Method of extractive summary with LexRank, for computing sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. Where the vectorizing (embedding) of each sentence is based on BERT by sentence-transformer.
- BESM (bert-extractive-summarizer method) : The sentence is embedding with BERT and centroid is obtained by K-Means method in the embedding vector space, and the sentence close to the centroid is selected by rank. (Derek Miller, 2019).
- BESM+koBERT (bert-extractive-summarizer + koBERT pre-trained model) : It becomes more optimized embedding for Korean.

Although it is common to apply the ROUGE matrix as a validation method for the summary algorithms, the ROUGE matrix could not be applied because the dataset with gold summary was not used in this experiment. Instead, we applied three perspectives: similarity, grammar, and isthmus resolution for performance comparison. In order to determine the isthmus resolution, the original document was divided into three part (introduction, body, and ending) and the variance of the similarity values of each part was calculated. The better the isthmus resolution is, the lower the variance will be.

As the validation document dataset, 100 test story documents were created by combining 30,000 Korean single sentences that had been used for learning Korean Grammar Discriminator in section3 part C. Each story document consisted of approximately 200 to 600 tokens and was divided into three parts: introduction, body, and conclusion. We compared and verified the previous three algorithms and our proposed method by measuring the average value for each performance comparison perspective.

C. *Performance comparison results*

Table1 shows the performance comparison results of the existing algorithm and our proposed method. 'Comp rate' means compression rate. Comp rate = Token count of summary text / token count of original document. Therefore, a lower 'Comp rate' means more compression. 'Intro', 'Body' and 'Ending' represent partial similarity of introduction, body, and conclusion part. 'Variance' represents the variance of the similarity of 'Intro', 'Body' and 'Ending'. In other words, a low value of 'Variance' means that isthmus is resolved. 'Total similarity' refers to the similarity compared to the original document, and finally 'Grammar' refers to the grammaticality measured by the grammar discriminator. The higher the value, the better the grammar, meaning that humans can read it naturally.

'Variance' came out the lowest on average despite the 'Comp rate' of the proposed method being the lowest. That is, it can be seen that the performance of the proposed method is good for resolving isthmus. In addition, 'Total similarity' was also higher than other algorithms. However, it showed the lowest results for 'Grammar'. This is probably because

the sentences recombined mechanically are more difficult to be natural than sentences composed by humans.

method	Comp rate	Partial similarity				Total similarity	Grammar
		Intro	Body	Ending	Variance		
SAM+WGAN	***0.1731***	0.5858	0.5466	0.5721	***0.0035***	***0.5652***	***0.9969***
BERT+LexRank	**0.1948**	0.3043	0.2891	0.3012	**0.0040**	**0.2941**	**0.9996**
BESM	**0.2763**	0.5095	0.4848	0.4535	**0.0068**	**0.4837**	**0.9999**
BESM+kobert	**0.2735**	0.5180	0.4868	0.4504	**0.0065**	**0.4873**	**0.9999**

Table 1 Performance comparison with existing summary algorithm, average value for 100 sample documents

V. CONCLUTION

The proposed method of this paper makes it possible to effectively abstract summary using GAN without a large summary data set. In addition, compared to the existing extraction method, it showed high similarity and low isthmus. But crucially, it's not yet flexible for humans to read. In order to develop into an algorithm that is used in the industrial field, a result that is smoother for humans to read will have to be produced. In addition, since it is unsupervised learning, it takes more computing time than abstract summarization algorithms such as Transformer, and memory efficiency is low because it is necessary to create an input of the same dimension as the number of tokens in the original document.

In order to solve such problems of grammar, processing time, and memory efficiency, it will be necessary to apply text generation through encoders and decoders, which have been previously studied. If unsupervised learning by GAN is combined with Transformer, it is expected that a more robust algorithm will be emerged, away from the existing just transformer method of supervised learning.

REFERENCES

[1] Hans Peter Luhn (1960). Keyword-in-context index for technical literature. American Documentation, 11(4):288–295. ISSN 0002-823

[2] Rada Mihalcea and Paul Tarau, (2004). TextRank: Bringing Order into Texts

[3] G¨une¸s Erkan. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization

[4] Susan T. Dumais (2005). "Latent Semantic Analysis". Annual Review of Information Science and Technology. 38: 188–230.

[5] Aria Haghighi, (2009). Exploring Content Models for Multi-Document Summarization

[6] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, Xuanjing Huang, (2020). Extractive Summarization as Text Matching

[7] Yang Liu and Mirella Lapata, (2019). Text Summarization with Pretrained Encoders

[8] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, (2014). Sequence to Sequence Learning with Neural Networks

[9] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation

[10] Colin Raffel. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

[11] Mike Lewis, (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

[12] Alec Radford, (2018). Language Models are Unsupervised Multitask Learners

[13] Guillaume Lample, (2019). Cross-lingual Language Model Pretraining

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, (2017). Improved Training of Wasserstein GANs

[15] Martin Arjovsky, Soumith Chintala, Léon Bottou, (2017). Wasserstein GAN

[16] Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu, (2016). Sequence Generative Adversarial Nets with Policy Gradient

[17] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, Dan Jurafsky, (2017). Adversarial Learning for Neural Dialogue Generation

[18] Nils Reimers and Iryna Gurevych, (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

[19] Sharma, P., & Li, Y. (2019). Self-Supervised Contextual Keyword and Keyphrase Retrieval with Self-Labelling

[20] Ian J. Goodfellow, (2014). Generative Adversarial Nets

[21] Yau-ShianWang, (2018). Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks

[22] Martin Arjovsky, (2017). Wasserstein GAN

[23] Derek Miller, (2019). Leveraging BERT for Extractive Text Summarization on Lectures