GAN based Korean summarizer using semi abstractive method

Hoon-Suk Lee
Department of Computer Engineering,
Dankook university
dolmani38@dankook.ac.kr
hoons1@asianaidt.co.kr

Soon-Hong An
Department of Computer Engineering
Dankook university
soonhoonan@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 as various transformer models 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 summary learning data set. An extractive summary method as a unsupervised learning can be used, but it has the disadvantage that an isthmus summary is made rather than the overall content. In order to overcome these obvious flaw, In this paper, we proposes a semi-abstractive method (SAM) that extracts words necessary for summary from the original text without a summary learning data set and generates a new summary by GAN algorithm. We show the distinction of this proposed method by demonstrating that it maintains the similarity of the overall contents of the document and the consistency of the grammar through experiments.**

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, the inshorts app, and many other news aggregator apps utilize text summarization algorithms. Let's look at the trend of the recent summary algorithm.

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

A. *Extractive text summary*

As a traditional method developed first, the main goal is to identify important sentences in the text and add them to the summary. The summary obtained is widely used in that it contains the exact sentence of the original text.

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. In this methods, key information is conveyed through the shortest possible text. In this approach, a new summary sentence is created, not the sentence extracted from the original text.

In this paper, as an abstractive text summary method, the original text is summarized by extracting the original text in terms of words, not sentences, and generating new sentences including the proper grammatical flow and contents of the original text.

II. RELATED WORK

First, looking at the existing 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. LSA (Susan T. Dumais, 2005) method was introduced in paper 'Potential Semantic Analysis'. This extracts semantic semantic sentences by applying SVD (Singular Value Decomposition) to the term document frequency matrix. KL-Sum (Aria Haghighi, 2009) was developed in 2009. A sentence is selected by calculating the KL-divergence of the word distribution in the original text and the word distribution by sentence.

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 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, such as matching. 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 Transformer and GAN techniques. In the next section, we describe a summary method proposed in this paper that implements a discriminator using a transformer and learns 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 method. Divide the original text by word and reassemble it to create several sentences that summarize the entire contents.

For this purpose, the Generator receives noise as much as the number of original words as input and has output as much as the number of original words. And then, the output corresponds to the probability value used for summarizing the words of each original text. Therefore, a summary sentence is generated by selecting a word corresponding to a specific probability value or a probability value of the highest ranking corresponding to the number of given words. Frame words are extracted from the original text 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 words, and then identifies whether the randomly generated sentences by the Korean grammar discriminator are grammatically human-readable sentences. Also, the similarity, which distinguishes the similarity of the original text, is adjusted so that the randomly generated sentences by the discriminator do not differ from the original text. The overall architecture is shown in the figure1 below.

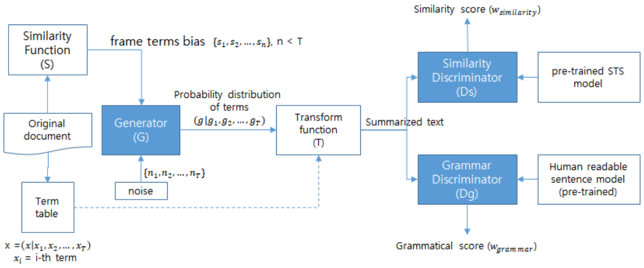
Similarity score ($w_{similarity}$)
Similarity Function (S)
frame terms bias $\{s_1, s_2, \ldots, s_n\}$, n < T
Similarity Discriminator (Ds)
pre-trained STS model
Probability distribution of terms
$(g|g_1, g_2, \ldots, g_T)$
Original document
Generator (G)
Transform function (T)
Summarized text
$\{n_1, n_2, \ldots, n_T\}$
noise
Term table
Grammar Discriminator (Dg)
Human readable sentence model (pre-trained)
x $=(x|x_1, x_2, \ldots, x_T)$
x_i = i-th term
Grammatical score ($w_{grammar}$)

Figure 1 Overall architecture of GAN based SAM for Korean

A. *Frame word extraction using similarity function*

Split the document into word units (the word unit here means spaces). When each word is called xi, the document can be expressed as follows.

$$x = \{x_1, x_2, \ldots, x_T\} \tag{1}$$

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

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

This is regarded as a continuous signal, and a word corresponding to the peak of the signal is extracted, and it is composed of frame words corresponding to the summary of the entire document.

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

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

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

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

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

However, the problem here is that even in SPD1~m, the peak is concentrated in the dominant word 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 SPD1 is subtracted to avoid the peak of the dominant word, and the frame corresponding to the story line can be extracted.

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

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 cosin 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 words corresponding to the total number of words in the original text are input as a bias. That is, there are two input terms and the output is equal to the number of words in the original text as it is the same as the input because we do not know which word will be used for the summary.

Random noise goes through several dense layers, but frame words 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 words, and a sentence summarizing the original text is generated. Also, appropriate words between each frame words are extracted from the original text by random noise. Then, in the generated probability distribution (Output), the word used in the summary is selected according to the following conditions.

$$t_j = \begin{cases} i & if\ g_i > \alpha \\ None & otherwise \end{cases} \qquad (8)$$

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 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.

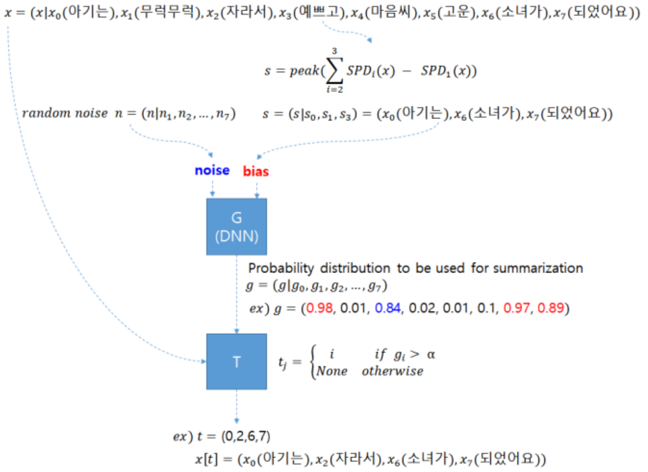
$x = (x|x_0$(아기는)$, x_1$(무럭무럭)$, x_2$(자라서)$, x_3$(예쁘고)$, x_4$(마음씨)$, x_5$(고운)$, x_6$(소녀가)$, x_7$(되었어요))
$s = peak(\sum_{i=2}^{3} SPD_i(x) - SPD_1(x))$
$random\ noise\ \ n = (n|n_1, n_2, \ldots, n_7)$
$s = (s|s_0, s_1, s_3) = (x_0$(아기는)$, x_6$(소녀가)$, x_7$(되었어요))
noise
bias
G (DNN)
Probability distribution to be used for summarization
$g = (g|g_0, g_1, g_2, \ldots, g_7)$
$ex)\ g =$ (0.98, 0.01, 0.84, 0.02, 0.01, 0.1, 0.97, 0.89)
T
$t_j = \begin{cases} i & if\ g_i > \alpha \\ None & otherwise \end{cases}$
$ex)\ t =$ (0,2,6,7)
$x[t] = (x_0$(아기는)$, x_2$(자라서)$, x_6$(소녀가)$, x_7$(되었어요))

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

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

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.

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

These problems are frequently encountered in the text generation algorithm through GAN, and in the paper (Yau-ShianWang at el,. 2018) 'Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks', 'Self-Critic Adversarial REINFORCE' is suggested. In this technique, a discrete sequence is supplied to the discriminator, so the slope of the discriminator cannot be directly back propagated to the generator. Here, the policy gradient method is used.

However, this method is possible when approaching the time series by RNN, 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.

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

The value of α above is a summary result and is determined by the number of words to be obtained. The larger the α value, the smaller the number of words in the summary result, and the smaller the α value, the larger the number of summary words.

An example of the output of T(G(z)) is as follows. (t|t$_0$,t$_1$,...t$_T$) ~ (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

$$G_{loss} = -\frac{1}{T}\sum_{i}^{T} w_D \cdot t_{i,G(z)} \tag{12}$$

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.

$$\min_{G}\max_{D_g,D_s} V\left(D_g,D_s,G\right) = E_{x\sim P_{data(x)}}[logD_g(x)] + E_{x\sim P_{data(x)}}[logD_s(x)] - \frac{1}{T}\sum_{i}^{T}(w_{D_g} + w_{D_s})t_{i,\,G(z)} \tag{13}$$

Each discriminator for measuring grammar and similarity was applied as a generalized delimiter using a pre-trained model. 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 text 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 because the Discriminator has already been learned by the pre-trained model. Therefore, the final objective function is simplified as follows.

$$\min_{G} V(G) = -\frac{1}{T}\sum_{i}^{T}(w_{D_g} + w_{D_s})t_{i,\,G(z)} \tag{14}$$

IV. Experiment

A. Experiment plan

Since the Korean abstractive summarizer does not currently have a comparable object, we compare it with the extractive summary method using the Korean BERT Model and verify the advantages of the proposed method.

We verify the superiority of the proposed method by comparing the following three methods and the proposed method for similarity and grammar.

- SAM+WGAN : proposed method
- BERT+LexRank : Method of extractive summary with LexRank after embedding each sentence 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.

However, in order to determine whether the entire story is included or not, the original text is divided into three part (introduction, body, and conclusion) and checked how it has similarity with each part. In addition, in order to verify the effectiveness of the proposed method according to the length of the document, we will compare a relatively short story with a long story. Table 1 below specifies the characteristics of each story.

Table 1 example stories to compare

	Contents	*Number of words*	*number of characters*
short story	*fairy and woodcutter story*	*128*	*546*
Long story	*Cinderella story*	*338*	*1,516*

B. WGAN learning

Training was conducted with 150 epochs, learning rate of 1e-4, and Adam optimizer. As can be seen from the figure3, it shows that the loss of similarity and grammar is lowered. That is, learning has progressed.

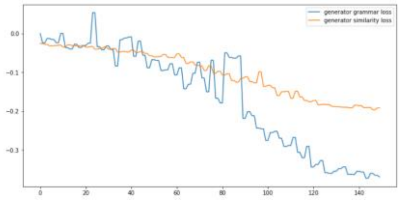

Figure 3 Similarity and grammatical loss according to epoch

C. Summary comparison results

Table 2 shows the comparison of results by summary method for relatively short stories. 'comp ratio' indicates the compression ratio of the summary. The lower the comp ration, the more the summary is compressed. For the reliability of the comparison verification, the comp ratio was similarly adjusted with other method. As can be seen from the table below, it shows that for relatively short stories (document), the effect of resolving isthmus compared to the extractive method does not appear much. As a result, there was no significant change in summarization similarity compared to the extractive method. However, from the comparison of the results of each summary method for the relatively long story in Table 3, it shows that the proposed method has the effect of resolving isthmus, although the compression ratio is the lowest compared to the extractive method. The similarities of the introduction, body, and conclusion are evenly distributed compared to other methods. In addition, in the distribution map of Fig. 4, it was confirmed that the deviation of the similarity between the introduction, the body, and the conclusion was the lowest in the proposed method.

method	Comp ratio	Similarity				grammar
		intro	body	end	total	
SAM+WGAN	***0.2364***	***0.8497***	***0.6673***	***0.5484***	***0.6832***	***0.9986***
BERT+LexRank	0.2309	0.2268	0.5041	0.6728	0.4769	0.9998
BESM	0.2709	0.9007	0.5679	0.7900	0.7066	0.9998
BESM+kobert	0.2291	0.9007	0.7162	0.4640	0.6993	0.9998

Table 2 Summary comparison results for relatively short story (fairy and woodcutter story)

method	Comp ratio	Similarity				grammar
		intro	body	end	total	
SAM+WGAN	***0.1646***	***0.5166***	***0.5829***	***0.6458***	***0.5766***	***0.9986***
BERT+LexRank	0.1508	0.2258	0.2443	0.1812	0.2302	0.9999
BESM	0.1632	0.4112	0.6119	0.5997	0.5620	0.9998
BESM+kobert	0.1632	0.4112	0.6119	0.5997	0.5620	0.9998

Table 3 Summary comparison results for relatively long story (Cinderella story)

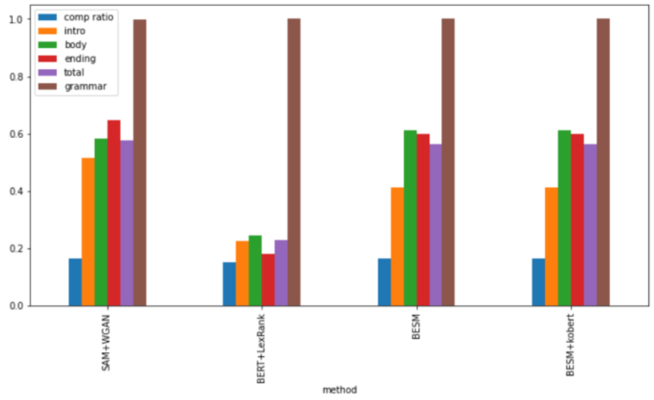

Figure 4 Summary comparison results for relatively long story

V. Conclution

The method proposed in this paper was able to solve the isthmus of the extractive method, and obtained the abstractive effect by the unsupervised learning of WGAN without the need for a large number of training sets required in the abstractive method. In terms of similarity compared to the original text, the results were comparable to the extractive method in a relatively short story, but the similarity was higher in the introduction, body, and conclusion than in extractive in a relatively long story.

However, the grammarlity was the lowest. This can be understood as a phenomenon that appears in the structure of a basic algorithm that generates words appropriate for grammar based on major words. Since it is a generator made with a general DNN structure, the grammar composition was not smooth. In the future, if the generator is configured with the seq2seq algorithm, it is expected that sentences with higher grammatical completion will be generated.

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

[7] 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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