LEARN TO ENCODE TEXT AS COMPREHENSIBLE SUMMARY BY GENERATIVE ADVERSARIAL NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Auto-encoders compress the input data into latent-space representation and reconstruct the original data from the representation. The latent representation cannot be easily interpreted by human. In this paper, we propose a new idea to train an auto-encoder that encodes input text into comprehensible sentences. The auto-encoder is composed of a generator and a reconstructor. The generator encodes the input text into a shorter word sequence, and the reconstructor recovers the input of generator from the output of generator. To make the generator output comprehensible by human, a discriminator restricts the output of the generator to look like the summaries written by human. By taking the generator output as the summary of the input text, abstractive summarization can be achieved without document-summery pairs as training data. Promising results were obtained on both English and Chinese corpora.

1 INTRODUCTION

When it comes to learning representation of data, one of the popular approaches is auto-encoder architecture which compresses the data into latent representation without supervision. This paper focuses on learning representation of text. Because text is a sequence of words, to encode a sequence, sequence-to-sequence (seq2seq) auto-encoder (Li et al., 2015; Kiros et al., 2015) is usually used, in which a RNN is used to encode the input sequence into a fixed length representation, and then another RNN to decode this input sequence out of that representation.

The latent representation can be used in the downstream applications, but they are usually not explanatory to human. Is it possible for machine to use human-readable language as latent representation of text? In this work, we show that it is absolutely possible for auto-encoder to use comprehensible natural language as latent representation of text if machine is given some example summaries written by humans. The comprehensible latent representation can be considered as the summaries of text. In this way, unsupervised abstractive summarization is achieved.

The idea of using human languages as latent representation has been explored (Miao & Blunsom, 2016), but only in semi-supervised scenario. The previous work uses a prior distribution from pretrained language model to constrain the generated sequence to natural language. However, they trained their model on some labeled data to teach their compressor network generate text summary. In this paper, we do not need any labeled data to learn such representation.

The proposed model is composed of three components: generator, discriminator and reconstructor. The generator and reconstructor together form an auto-encoder of text. The generator generates latent representation from input text as encoder. Instead of using a vector as latent representation, the generator generates a word sequence much shorter than the input text. From the shorter text, the reconstrutor generates the original input of the generator. By minimizing the reconstruction errors, the generator learns to generate short text containing the main information of the original input. We choose seq2seq model as the models of generator and reconstructor because both have input and output sequences with different lengths[1].

However, it is very possible that the output word sequence of generator can be processed by the reconstructor, but not comprehensible by humans. In this paper, instead of regularizing the generator output by pretraining language model (Miao & Blunsom, 2016), we use the third component, discriminator, to regularize the output word sequence of generator. The discriminator and the gener-

[1]The auto-encoder composed of the generator and reconstructor is thus a seq2seq2seq model.

ator form a generative adversarial network (GAN) (Goodfellow et al., 2014). GANs are generative models composed of a generator and a discriminator. The discriminator tries to discriminate between the output of generator and real data, and the generator produces outputs as similar as real data as possible to confuse discriminator. In this paper, we feed document summaries to the discriminator as real data[2]. The summaries do not have to be paired with the documents used to train generator and reconstructor. With GAN framework, discriminator teaches generator how to create human-like summary sentences as latent representation, but it only guarantees the generator to generate grammatically correct sentences but not necessarily represent the input text. On the other hand, reconstructor teaches generator how to produce sentence capturing the core idea of source text.

However, generating discrete distribution with GAN is extremely challenging, since it is difficult to evaluate the distance between continuous distribution from generator and discrete distribution of real sample. In addition, if we feed sampled words from generator output distribution to discriminator, the process of word selection is non-differentiable, which makes gradient from discriminator unable to back-propagate to generator. There are two kinds of ways on language generation with GAN: training with adversarial REINFORCE regarding words as actions or directly feeding output layer of generator to discriminator making gradient able to back-propagate to generator. In this work, on language generation with GAN, we explore both methods in the experiments.

We evaluate the results on abstractive text summarization task in which machine generates text summary by its own words. The model is learned from a set of unpaired documents and summaries. The summaries can come from another set of documents not related to the training documents. Hence, the training is unsupervised. We take the output word sequence of the generator as the summaries of the input text. The results show that the generator can generate summaries with reasonable quality on both English and Chinese corpora.

2 Related Work

GAN on Language Generation

The major challenge for applying GAN on sentence generation is the discrete nature of natural language. To generate a word sequence, the generator usually has non-differential part like $argmax$ or other sample function, which makes the original GAN fails. Therefore, new kinds of GANs are proposed for sentence generation.

SeqGAN (Yu et al., 2017) tackles the sequence generation problem with reinforcement learning. In this paper, we refer this kind of approaches as adversarial REINFORCE. In adversarial REINFORCE, the generator is regarded as agent, the generated sequence of words is viewed as sequence of actions, and the current state is defined on generated sequence so far and input prior. However, the discriminator only measures the quality of whole sentence, and thus the reward is extremely sparse and the rewards assigned to all actions in sequence are all the same. To tackle this problem, they propose MC search to evaluate approximate rewards at each timestep, but this method suffers from high time complexity. Following this idea, (Li et al., 2017) uses another way to evaluate expected reward at each timestep. They break both generated sequence and real sequence into partial sequence, and discriminator discriminates between generated and real partial sequence. Inspiring by this idea, we introduce another way to evaluate expected reward at each timestep in our proposed self-critic in adversarial REINFORCE.

On the other hand, in Improved Training of Wasserstein GANs paper (Gulrajani et al., 2017), instead of feeding discrete word sequence, they directly feed generator output layer to discriminator. The main reason this method works is that they use earth mover's distance on GAN proposed in (Arjovsky et al., 2017) which is capable to evaluate the distance between discrete and continuous distribution. In order to satisfy the requirement of earth mover's distance, they use gradient penalty trick to confine complexity of discriminator function. Their method achieves an amazing result that it's the first paper training GAN on language generation without pretraining. In our work, we also make experiment on this method with discriminator setting almost same as original paper.

[2]Instead of using general sentences as real data for discriminator, we choose to use sentences from summaries here because they have their own unique distribution. For example, the structure of summary sentence is concise, and they usually contain some specific words like country or human name, or some key verbs in source text. Because the following application is document summarization, using document summaries as real data is very helpful. It is possible to use text with different styles as real data for discriminator to make the output of the generator have different styles.

ABSTRACTIVE TEXT SUMMARIZATION

Recently, the model architectures of abstractive text summarization basically use the sequence-to-sequence (Sutskever et al., 2014) framework combined with some extra novel mechanism. One popular mechanism is attention (Bahdanau et al., 2015), which has shown to be helpful on summarization (Nallapati et al., 2016; Rush et al., 2015). It's also possible to directly optimize evaluation metric such as ROUGE (Lin, 2004) with reinforcement learning (Ranzato et al., 2016; Paulus et al., 2017; Bahdanau et al., 2016). Hybrid pointer-generator network (See et al., 2017) selects words from original text with pointer (Vinyals et al., 2015) or from the whole vocabulary with a trained weight. In order to eliminate repetition, it uses coverage vector (Tu et al., 2016) keeping track of attended words and use coverage loss to encourage model focus on diverse words. While most papers focus on supervised learning with novel mechanism, we explore on training model with unsupervised manner.

3 PROPOSED METHOD

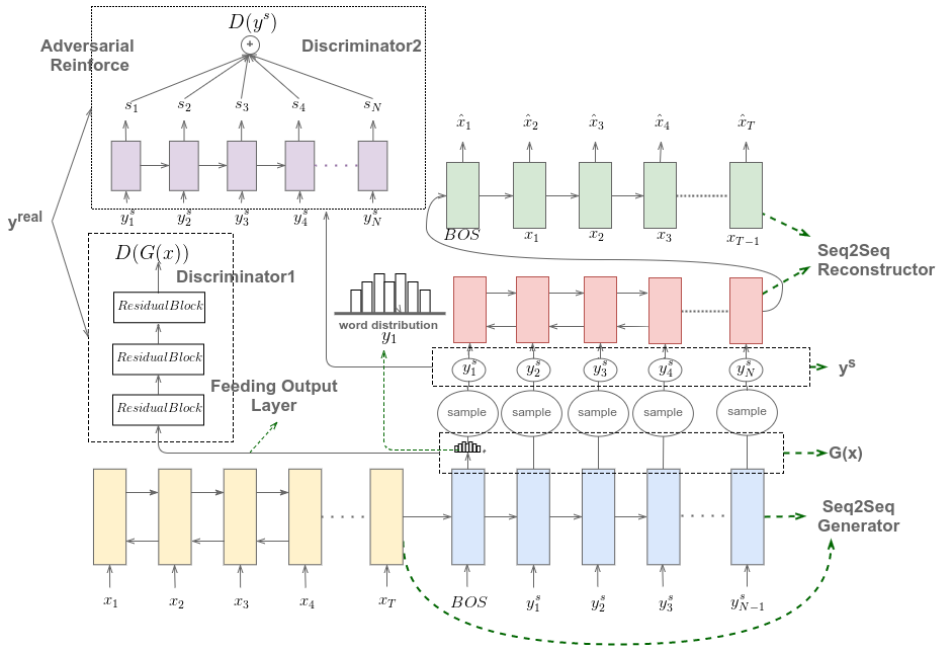

Figure 1: The overview of the proposed model. The generator network and reconstructor network are seq2seq pointer-generator network, but for simplicity, we omit the pointer and the attention part.

The overview of the proposed model is shown in figure 1. The model is composed of three components: generator G, discriminator D and reconstructor R. Both G and R are seq2seq hybrid pointer-generator network (See et al., 2017). They take a word sequence as input and output a sequence of word distributions. On the other hand, D takes a sequence as input and outputs a scalar. The model is learned from a set of documents x and summaries y^{real}, but *the documents and summaries are unpaired*.

To train the model, a training document $x = \{x_1, x_2, ..., x_t, ..., x_T\}$, where x_t represents a word, is fed to G, and it outputs a sequence of word distributions $G(x) = \{y_1, y_2, ..., y_n, ..., y_N\}$, where y_n is a distribution over all the words in the lexicon. Then we sample a word y_n^s from each distribution y_n, and a word sequence $y^s = \{y_1^s, y_2^s, ..., y_N^s\}$ is obtained according to $G(x)$. We feed the sampled word sequence y^s to reconstuctor R, and R outputs another sequence of word distributions $\hat{x}$. The reconstuctor R manages to reconstruct the original text x from y^s. That is, we want the output of reconstuctor $\hat{x}$ as close to the original text x as possible, so the loss for training the reconstuctor R, R_{loss}, is defined as below.

$$R_{loss} = \sum_{k=1}^{K} l_s(x, \hat{x}), \tag{1}$$

where reconstruction loss $l_s(x, \hat{x})$ is cross-entropy loss computed between reconstructor output sequence $\hat{x}$ and source text x. The subscript $_s$ in $l_s(x, \hat{x})$ means that $\hat{x}$ is reconstructed from y^s. K is the number of training examples (documents), and (1) is the summation of the cross-entropy loss over all the training documents x.

In the proposed model, the generator G and reconstuctor R form an auto-encoder. However, the reconstuctor R does not directly take generator output distribution $G(x)$ as input [3]. Instead, the reconstuctor takes sampled discrete sequence y^s as input. Due to the non-differentiable property of discrete sequence, we apply the REINFORCE algorithm, which will be described in section 4.

In addition to reconstruction, we need discriminator D to discriminate between the real summary sequence y^{real} and the generated sequence y^s to regularize generated sequence satisfying summary distribution. D learns to give y^{real} larger scores while y^s smaller scores. The loss for training the discriminator D is denoted as D_{loss}, which will be further described in section 5.

G learns to minimize the reconstruction error R_{loss}, whereas maximize the loss of the discriminator D to generate summary sequence y^s that can not be tell real or generated by D. The loss for training the generator G, G_{loss}, is shown as below:

$$G_{loss} = \alpha R_{loss} - D'_{loss}, \tag{2}$$

where D'_{loss} is highly related to D_{loss}, but not necessary the same, and α is a hyper-parameter. After obtaining the optimal generator G^* by minimizing (2), it can be used to generate summaries.

The the generator G and discriminator D form a GAN. We use two different adversarial training methods to train D and G, and these two methods have their own discriminators 1 and 2 as shown in figure 1. Discriminator 1 takes generator output layer $G(x)$ as input, while discriminator 2 takes sampled discrete word sequence y^s as input. The two methods will be described respectively in sections 5.1 and 5.2.

4 Minimizing Reconstruction Error

Due to the non-differentiable property of discrete sequence, we apply REINFORCE algorithm. The generator is considered as an agent whose reward given source text x is $-l_s(x, \hat{x})$. Maximizing the reward is equivalent to minimizing the reconstruction loss R_{loss} in (1). However, the reconstruction loss differs a lot from sample to sample, and thus the rewards to generator is not stable. We should add a baseline to reduce their difference. We apply self-critical sequence training (Rennie et al., 2017), and the modified reward $r^R(x)$ from reconstructor R with baseline for generator is shown as below,

$$r^R(x) = -l_s(x, \hat{x}) - (-l_a(x, \hat{x}) - b).$$

where $-l_a(x, y^a) - b$ is the baseline. $l_a(x, \hat{x})$ is also the cross-entropy reconstruction loss as $l_s(x, \hat{x})$, except that $\hat{x}$ is obtained from y^a instead of y^s. y^a is a word sequence $\{y_1^a, y_2^a, ..., y_n^a, ..., y_N^a\}$, where y_n^a is selected by $argmax$ function from the output distribution of generator y_n. Because in early stage of training sequence y^s barely gets higher reward than sequence y^a, we introduce the second baseline score b gradually decreases to zero to encourage exploration. Then, the generator is updated by REINFORCE algorithm with reward $r^R(x)$ to minimize R_{loss}.

5 GAN Training

We feed summary sentence to discriminator as real sample, with adversarial training, generator manages to produce sentence as similar to summary as possible. In this paper, we make experiment on two kinds of methods on language generation with GAN. In section 5.1 we directly feed generator output probability distributions to discriminator and apply Wasserstein GAN (WGAN) with gradient penalty. In section 5.2, we explore adversarial REINFORCE that feeds sampled discrete word sequence to discriminator and uses the evaluation of quality of sequence from discriminator as reward signal to generator.

[3] We found that if the reconstuctor R directly takes $G(x)$ as input, the generator G would learn to put the information of the input text in the distribution of $G(x)$, and thus it would be hard to sample meaningful sentences from $G(x)$.

5.1 DISCRIMINATOR 1: WASSERSTEIN GAN

The model of discriminator is shown as **discriminator1** in lower-left of figure 1. The loss D_{loss} of discriminator is:

$$D_{loss} = \frac{1}{K}\sum_{k=1}^{K} D(y^{s(k)}) - \frac{1}{K}\sum_{k=1}^{K} D(y^{real(k)}) + \beta(\Delta_{y^{i(k)}} D(y^{i(k)}) - 1)^2, \tag{3}$$

where K denoted the number of training examples, and k denoted k'th example in batch. The last term in (3) is gradient penalty (Gulrajani et al., 2017). We interpolate between generator output layer $G(x)$ and real sample y^{real}, and apply gradient penalty on interpolated sequence y^i. β decides the scale of gradient penalty.

In equation (2), we add the reconstruction loss for generator R_{loss} and loss related to discriminator D'_{loss} together to formulate total generator loss. For WGAN, D'_{loss} is the score of the generated example,

$$D'_{loss} = \frac{1}{K}\sum_{k=1}^{K} D(G(x^{(k)}))$$

5.2 SELF-CRITIC ADVERSARIAL REINFORCE

In this section, we describe in detail our proposed adversarial REINFORCE method. The core idea is we make LSTM discriminator evaluate the current quality of generated sequence $\{y_1^s, y_2^s, ..., y_i^s\}$ at each time step i. Hence, generator will know that compared to last time step, the generated sentence becomes worse or better, so it can easily find the problematic generation step in a long sequence, and fix the problem easily.

5.2.1 DISCRIMINATOR 2

As **discriminator2** shown in figure 1, the model of discriminator is a one-way LSTM network which takes discrete word sequence as input. At time step i, given input word y_i^s it will predict the current score s_i based on the sequence $\{y_1, y_2, ..., y_i\}$. The score is viewed as the quality of current sequence.

In order to compute the loss of discriminator, we sum all the scores $\{s_1, s_2, ..., s_N\}$ of the whole sequence y^s and the loss of discriminator D_{loss} is:

$$D(y^s) = \frac{1}{N}\sum_{n=1}^{N} s_n$$

$$D_{loss} = \frac{1}{K}\sum_{k=1}^{K} D(y^{s(k)}) - \frac{1}{K}\sum_{k=1}^{K} D(y^{real(k)})$$

where K and N denote to the number of training examples and generated sequence length respectively. With the loss mentioned above, the discriminator tries to determine whether the current sequence is real or false as early as possible. The earlier timestep discriminator determines current sequence is real or false, the lower loss it get. We show a real example on how it works in figure 2.

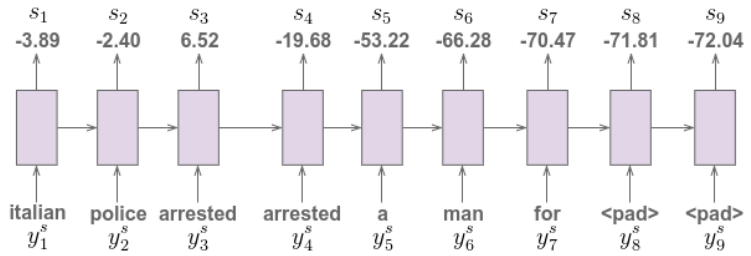

Figure 2: When the second "arrested" appears, the discriminator determines this example comes from generator. Hence, after this timestep, it outputs the scores as low as possible

5.2.2 SELF-CRITICAL GENERATOR

Since we feed discrete sequence y^s to discriminator, the gradient from discriminator can not directly back-propagate to generator. Here, we apply policy gradient method. At timestep i, we use the $i-1$ timestep score s_{i-1} from discriminator as its self critical baseline. The reward r_i^D evaluates compared to timestep $i-1$ the quality of sequence in timestep i is worse or better. The generator reward r_i^D from D is:

$$r_i^D = \begin{cases} s_i & \text{if i = 1} \\ s_i - s_{i-1} & \text{otherwise.} \end{cases}$$

However, some sentence may be judged as bad sentence at previous timestep, but judged as good sentence at latter and vice versa. To tackle this problem, we use discounted expected reward d with discounted factor γ, the discounted reward d_i at time step i is:

$$d_i = \sum_{j=i}^{N} \gamma^{j-i} r_j^D$$

The adversarial REINFORCE score related to discriminator D'_{loss} in equation (2) is:

$$D'_{loss} = E_{y_i^s \sim p(y_i^s | y_i^s, \ldots, y_{i-1}^s, x)}[d_i]$$

We apply likelihood ratio trick to approximate gradient.

6 EXPERIMENT

6.1 EXPERIMENTAL SETUP

Our model is evaluated on two datasets: Chinese Gigaword and English Gigaword. The seq2seq models of generator and reconstructor are all composed of two one-layer LSTM with hidden layer size 600. In section 5.1 which feeds generator output layer to discriminator, we used 4 residual block with hidden dimension 512 as our discriminator, set the weight β of gradient penalty to 10, and used RMSPropOptimizer with learning rate 0.00001 and 0.001 on generator and discriminator respectively. In section 5.2, we applied weight clipping to approximate function required by Wasserstein distance. We only used one hidden layer one-way LSTM with hidden size 512 and clipped the value of its weights to ± 0.15, and used RMSPropOptimizer with learning rate 0.0001 and 0.001 on generator and discriminator respectively. During testing, when using the generator to generate summary, we simply took the words with highest probability without beam-search, and we eliminated repetition.

Before jointly training whole model, we pretrained three major components, generator, discriminator and decoder, in our model separately. First, we pretrained generator with unsupervised approach to make generator grasp some semantic meaning of source text, and roughly initialized with a language model. We found that the different pretraining methods on generator influenced final performance dramatically in all of our experiment, and thus it's crucial to find a proper unsupervised pretraining method to help machine understand semantic meaning. We use different pretraining strategies which will be further described in the following discussion. We pretrained discriminator and decoder respectively with pretrained generator's outputs to ensure these two critic networks provide good feedback to generator.

6.2 CHINESE GIGA WORD

Methods		ROUGE-1	ROUGE-2	ROUGE-L
(A) Training with Paired Data (Supervised)		48.664	33.907	45.685
(B) Trivial Baselines: lead - 15		30.077	18.237	27.736
(C) Unsupervised	(B-1) Pretrained Generator	28.122	16.656	26.227
	(C-2) **WGAN**	37.803	24.460	35.116
	(C-3) **Adversarial REINFORCE**	40.053	26.126	37.118

Table 1: Result on Chinese Gigaword. In supervised training, we trained our model with labeled data and with coverage loss mentioned in (See et al., 2017).

The Chinese Gigaword corpus is composed of 2.2M paired data of headlines and news. We preprocessed the raw data as following. First, we selected the most frequent 4K Chinese characters as our

Source Text: three stores and markets in beijing 's fengtai district have been forced to shut down and yesterday each was fined ###,### yuan -lrb- ##,### us dollars -rrb- for violating laws and regulations on fire prevention and control .	
Ground Truth: stores markets punished for lack of fire controls	**(A)Supervised Result:** beijing 's district stores shut down
(C-1)Pretrained Generator: and yesterday prevention have been forced to shut down	**(D-1)Pretrained Generator:** three stores and markets fined ###,### yuan dollars
(C-2)WGAN: three stores markets in beijing 's district have forced to shut down yesterday	**(C-3)Adversarial REINFORCE:** three stores in beijing forced to shut down for violating regulations
(E-1)WGAN : three stores and markets was forced to shut down	**(E-2)Adversarial REINFORCE:** three stores and markets was forced to shut down and yesterday each was fined

Figure 3: Real example from our model in English Gigaword. Our methods is capable to generate summary capturing the core idea of article.

Source Text: italian prime minister silvio berlusconi arrived monday in the libyan capital tripoli , one of only a few high-ranking european officials to visit the country in the last \#\# years .	
Ground Truth: italy 's prime minister arrives in libya ; highest-ranking european to visit in \#\# years	**(A)Supervised Result:** berlusconi arrives in libya
(C-1)Pretrained Generator: the libyan capital in tripoli to visit the country	**(D-1)Pretrained Generator:** , has been trying to visit the country from tripoli
(C-2)WGAN: italian visit in libyan capital tripoli	**(C-3)Adversarial REINFORCE:** italian prime minister to visit berlusconi in tripoli years
(E-1)WGAN : italian prime minister silvio berlusconi arrived in libyan capital tripoli	**(E-2)Adversarial REINFORCE:** arrived in the libyan capital tripoli

Figure 4: In part (C-3), some words in summary sentences are arranged in incorrect order.

vocabulary. We filtered out headline-news pairs containing too many Chinese characters out from our vocabulary or the length of news was extremely long or short. After filtering, we obtained 1.1M headline-news pairs. We randomly selected 5K headline-news pairs from filtered data as our testing set, 5K headline-news pairs as our validation set and the remaining pairs as our training set. During training or testing, the generator only took the first 80 Chinese characters of the source texts as input.

Before jointly training whole model, we pretrained generator to predict the next sentence of part of source text. The next sentence was not included in given text to generator. If the next sentence of given text contained more than 50% words not appearing in given text, we filtered out this pretraining sample pair. This pretraining method allows generator to capture the important semantic meaning of source text and also initialize with a crude language model.

The results are shown in table 1. Row (A) is the results using 1.1 million document-summary pairs to directly train the generator without reconstructor and discriminator, which is the upper bound of the proposed approach. In row (B), we simply took the first fifteen words in a document as summary[4]. Part (C) are the results obtained in the unsupervised scenario without paired data. We show the results of the pre-trained generator in row (C-1), and rows (C-2) and (C-3) are the results for two GAN training methods respectively. We found that although there is still a performance gap between the unsupervised methods and supervised methods (rows (C-2), (C-3) v.s. (A)), the

[4]The number of words is chosen to optimize the evaluation metrics.

proposed method obtained much better performance than the trivial baselines (rows (C-2), (C-3) v.s. (B)). In the above experiments, we set the weight α in 2 controlling R_{loss} to be 10.

6.3 English Giga Word

Methods		ROUGE-1	ROUGE-2	ROUGE-L
(A) Training with Paired Data(Supervised)		37.469	16.272	35.175
(B) Trivial Baselines - lead 8		27.663	10.246	25.852
(C) Unsupervised	(C-1) Pretrained Generator	21.269	5.608	18.896
	(C-2) **WGAN**	33.043	12.222	30.121
	(C-3) **Adversarial REINFORCE**	32.826	9.332	28.727
(D) Transfer Learning (Pretrain)	(D-1) Pretrained Generator	22.540	6.652	20.879
	(D-2) **WGAN**	32.405	12.313	29.689
	(D-3) **Adversarial REINFORCE**	31.487	10.495	28.248
(E) Transfer Learning (Pretrain+Title)	(E-1) **WGAN**	29.912	10.695	27.324
	(E-2) **Adversarial REINFORCE**	27.755	9.280	24.860

Table 2: Result on English Giga Word: In row (B), we selected the first eight words of article as summary. In transfer learning experiment, in part (D), we pretrained generator on CNN/Diary. While in part (E), we not only pretrained on CNN/Diary but also used title from CNN/Diary as real data for discriminator.

On English Gigaword, we set our vocabulary size to 15k, and we used the dataset preprocessed by (Rush et al., 2015) for training and testing. We used 3.8M unpaired training data for our training, and we used full 200k filtered data in validation set for testing.

The length of source texts in English Giga Word dataset is comparatively short, and it's difficult to split the last sentence from source text, hence the previous pretraining method on Chinese Gigaword does not fit for this dataset. In order to get a proper initialization, we randomly selected consecutive 6-11 words in source text, then we randomly swapped 70 % words in source text. Given text with incorrect words arrangement, generator predicted the selected words in correct arrangement. The reason we pretrained with this approach is that we expect the generator to initialize with a rough language model and focus on diverse part of text[5].

The results on English Giga Word are shown in table 2. Compared with the trivial baselines (row (B)), the proposed approach (row (C-2)and (C-3)) improved a lot in terms of ROUGE-1.As shown in figure 3, unsupervised method is capable to select key words in source text and generate text summary. However, as shown in table 2, although both unsupervised methods get ROUGE-1 score close to supervised training, they get lower score on ROUGE-2 especially on training GAN with reinforcement learning. The main reason is they are able to extract key words from source text, but sometimes fail to arrange these words in correct order. As shown in figure 4, the words in sentences generated by unsupervised manner are put in incorrect arrangement. In part (C-3), the words are arranged in incorrect semantic meaning that the Italian prime minister, Berlusconi, should not visit himself.

6.4 Transfer Learning

In this subsection, we study transfer learning. We used CNN/Daily Mail dataset Hermann et al. 2015; Nallapati et al. 2016 preprocessed by the script provided by See et al. as our source domain S, and English Giga Word as our target domain T.

The data distributions among two datasets are quite different. In English Giga Word, the articles consist of 32 words in average and the summary consists of one sentence with 8 words in average. While in CNN/Daily Mail dataset the articles are composed of 790 words and the summary are composed of several sentences. We preprocessed the source data into three parts: 50K source articles S_a, 50K source summaries S_t and 240K source real sentences S_r. We only took first 30-45 words in original source domain articles as our new source article S_a. However, unlike English Giga Word, the summary in S is composed of few sentences. We split the summary into several sentences and viewed all split sentences as ground truth S_t to S_a. Hence, one S_a has several ground truth S_t. Then,

[5]However, in fact, this is not a proper initialization, in Chinese Giga word we also made experiment on pretraining with this manner, but the performance of previous method used on Chinese Gigaword is much better.

all the sentences in original source summary were viewed as source real summary sentences S_r. We filtered out the samples containing so many unknown words or the bad summary sentences.

The transfer learning is applied on two direction. The first direction is we pretrained our generator on the data from source dataset. We pretrained our generator with S_a as input, and generator predicted S_t. We pretrained generator in this manner in all of the transfer learning experiments. Then, after pretraining, the generator was further learned jointly with reconstructor and discriminator on the data from target domain. The results are shown in the part (D) of Table 2. We found that pretraining on the source data set did not degrade the performance, and even improved the performance in some cases (parts (D) v.s. (C)).

The second direction is we used the summary from source domain as our real data. The experiments we made so far required unpaired data of summary and text in target domain. In this experiment, we made our task more challenging that we used summary S_r from source domain S as real data for discriminator, and generator took text of target domain as input.

In fact, each summarization task dataset has it own data distribution, for example writing style or preferred words of summary or content of articles. In this experiment, we set the weight $alpha$ to be 50. We used larger α to prevent overfitting to S_r. If the weight α is too small, as training going on, the summary produced by generator becomes more and more unrelated to article, and ROUGE score becomes lower and lower.

The results without using summaries in the target domain are shown in the part (E). We found that using title from another dataset gets lower ROUGE score on target testing set (parts (E) v.s. (D)) due to the mismatch between the summaries of the source and target domains. However, the discriminator still roughly regularizes the language model of generated word sequence. After training, the model still greatly enhanced ROUGE score of the pre-trained model (rows (E-1), (E-2) v.s. (D-1)). The results in part (E) are just comparable with the trivial baselines in row (B), but from the real examples, we found that the results in part (E) were better, which is not reflected in the ROUGE scores. In figure 3, the result in part (E) is better than leading 8 words and the result of pretrained generator in part (D-1). To support this idea, we provide more examples in appendix.

6.5 GAN Training

The two methods for training GAN are not comparable since their settings are almost different, but we can still discuss the pros and cons of these two methods. In fact, both methods are sensitive to initialization of generator especially for adversarial REINFORCE, since it requires a good exploration direction. In Chinese Gigaword, as shown in table 1, since generator was initialized with a better parameters, the adversarial REINFORCE got better performance than its performance in English Giga Word. In addition, compared with English Gigaword, the vocabulary size of Chinese Gigaword is smaller, which means it's easier for generator to select a word as action. On the other hand, training with feeding output layer converges faster. It manages to make the distribution as sharp as possible at early stage of training because it directly evaluates the distance between continuous distribution from generator and discrete distribution of real data.

7 Conclusion and Future Work

By GAN, we propose a model to encode text as comprehensible summary, which is learned without document-summary pairs. Promising results were obtained on both Chinese and English corpora. In future work, we are going to explore more techniques for natural language generation by GAN. Moreover, we will use extra discriminators to control the style and sentiment of the generated summaries.

References

Martin Arjovsky, Soumith Chintala, and Lon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*, 2016.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

Karl Moritz Hermann, Tom Koisk, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *NIPS*, 2015.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. *NIPS*, 2015.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *ACL*, 2015.

Jiwei Li, Will Monroe, Tianlin Shi, Sbastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *In Text summarization branches out: ACL workshop*, 2004.

Yishu Miao and Phil Blunsom. Language as a latent variable: Discrete generative models for sentence compression. *EMNLP*, 2016.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. *EMNLP*, 2016.

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

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *ICLR*, 2016.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *CVPR*, 2017.

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

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

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *NIPS*, 2014.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. *ACL*, 2016.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *NIPS*, 2015.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. *AAAI*, 2017.

Source Text: former zambian president kenneth kaunda appeared in court monday on charges of holding an illegal rally , declaring that he would continue to fight the `` oppressive regime '' of president fredrick chiluba .	
Ground Truth: former zambian president in court for illegal assembly	**(A)Supervised Result:** kaunda to continue to fight chiluba
(C-1)Pretrained Generator: president kenneth chiluba appeared in court monday on charges of	**(D-1)Pretrained Generator:** kaunda declaring that he would continue to fight the regime says
(C-2)WGAN: zambian kenneth kaunda in court charges of holding illegal rally	**(C-3)Adversarial REINFORCE:** former zambian president to continue illegal rally fight
(E-1)WGAN : former zambian president kenneth kaunda appeared in court of illegal rally	**(E-2)Adversarial REINFORCE:** he appeared in court he would continue to fight the regime

Figure 5: In part (E-2), the summary sentence begins with word ”he”, which never appears in English Gigaword summary sentences.

Source Text: hong kong tourist association -lrb- hkta -rrb- said wednesday it regretted having placed an advertisement thanking sponsors of last week 's lunar new year parade which killed one man and left ## others injured .	
Ground Truth: hong kong tourist body expresses regret over advertisement	**(A)Supervised Result:** hong kong tourist regrets having placed advertisement sponsors
(C-1)Pretrained Generator: hong kong said wednesday it regretted an advertisement	**(D-1)Pretrained Generator:** one man was killed in the head of the sponsors of the said
(C-2)WGAN: hong kong tourist association regretted having placed advertisement sponsors	**(C-3)Adversarial REINFORCE:** hong kong tourist killed advertisement sponsors of last year 's lunar parade
(E-1)WGAN : hong kong tourist association regretted having placed an advertisement sponsors	**(E-2)Adversarial REINFORCE:** it regretted having placed advertisement sponsors of last week 's lunar new year

Figure 6: The grammar of sentence in part (C-3) is correct, but the semantics of it is incorrect

Source Text: experts from iran and the un nuclear watchdog met thursday in vienna to discuss tehran 's plans to resume atomic fuel research , an official from the agency said .	
Ground Truth: iranian experts meet with un nuclear watchdog	**(A)Supervised Result:** iran un nuclear watchdog discuss tehran
(C-1)Pretrained Generator: to discuss tehran 's nuclear research watchdog	**(D-1)Pretrained Generator:** . official says experts from iran 's president 's watchdog met
(C-2)WGAN: experts iran to un nuclear watchdog in vienna	**(C-3)Adversarial REINFORCE:** experts discuss un nuclear watchdog plans to resume tehran
(E-1)WGAN : experts from iran and the un nuclear watchdog met	**(E-2)Adversarial REINFORCE:** experts from iran and the un nuclear watchdog met

Figure 7

Source Text: european shares fell monday , pressured by higher crude prices after oil giant bp said it will shut down a key production field and on caution before a u.s. interest-rate decision .	
Ground Truth: european stocks end lower	**(A)Supervised Result:** european stocks fall on bp decision
(C-1)Pretrained Generator: prices fell monday after caution on caution	**(D-1)Pretrained Generator:** oil giant bp says higher crude prices on caution
(C-2)WGAN: european shares shut down higher after oil	**(C-3)Adversarial REINFORCE:** european shares shut down after higher oil production
(E-1)WGAN : european shares fell by higher crude prices after oil	**(E-2)Adversarial REINFORCE:** by higher crude prices after oil giant and on caution before it will shut

Figure 8

Source Text: russian foreign minister igor ivanov urged iran to be open about its nuclear programs during a meeting with his iranian counterpart at the united nations , the foreign ministry said tuesday .	
Ground Truth: russian minister urges iran to be open about nuclear programs	**(A)Supervised Result:** russian fm urges iran to be open about nuclear programs
(C-1)Pretrained Generator: ivanov urged iran to be open its nuclear programs during	**(D-1)Pretrained Generator:** russian foreign minister igor ivanov says meeting will be open
(C-2)WGAN: russian igor to be open about nuclear programs during meeting	**(C-3)Adversarial REINFORCE:** russian foreign minister to open iran meeting about its nuclear programs
(E-1)WGAN : russian foreign minister igor ivanov urged iran to be open about nuclear programs	**(E-2)Adversarial REINFORCE:** new foreign feature urges iran to be open about its nuclear programs

Figure 9

Source Text: the bewildering fight between the government and telemarketers over the national do-not-call list took another turn when a second federal agency said it would enforce the program , promising that consumers would soon see some reduction in telephone sales pitches .	
Ground Truth: fcc steps in to enforce do-not-call list ; bush signs new law to support program	**(A)Supervised Result:** fight against consumers
(C-1)Pretrained Generator: the second list in telephone sales that would enforce	**(D-1)Pretrained Generator:** , promising that consumers would enforce turn
(C-2)WGAN: fight between government over national list took turn	**(C-3)Adversarial REINFORCE:** fight between government pitches see another reduction
(E-1)WGAN : the fight between the government and over the national list took another turn	**(E-2)Adversarial REINFORCE:** fight between government and over national list took another turn when a second federal

Figure 10