Sequence Generative Adversarial Network for Long Text Summarization

Hao Xu
School of Cyber Security
University of Chinese Academy of Sciences
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
xuhao2@iie.ac.cn

Yanan Cao
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
caoyaonan@iie.ac.cn

Ruipeng Jia
School of Cyber Security
University of Chinese Academy of Sciences
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
jiaruipeng@iie.ac.cn

Yanbing Liu
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
liuyanbing@iie.ac.cn

Jianlong Tan
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
tanjianlong@iie.ac.cn

***Abstract*—In this paper, we propose a new adversarial training framework for text summarization task. Although sequence-to-sequence models have achieved state-of-the-art performance in abstractive summarization, the training strategy (MLE) suffers from exposure bias in the inference stage. This discrepancy between training and inference makes generated summaries less coherent and accuracy, which is more prominent in summarizing long articles. To address this issue, we model abstractive summarization using Generative Adversarial Network (GAN), aiming to minimize the gap between generated summaries and the ground-truth ones. This framework consists of two models: a generator that generates summaries, a discriminator that evaluates generated summaries. Reinforcement learning (RL) strategy is used to guarantee the co-training of generator and discriminator. Besides, motivated by the nature of summarization task, we design a novel Triple-RNNs discriminator, and extend the off-the-shelf generator by appending encoder and decoder with attention mechanism. Experimental results showed that our model significantly outperforms the state-of-the-art models, especially on long text corpus.**

***Index Terms*—Sequence Generative Adversarial Network, Text Summarization, Deep learning, Reinforcement learning**

I. Introduction

With the dramatic growth of data, more and more fragmented information overwhelms readers, in the form of e-Newspapers, technical reports and tweets etc. Due to the expensive cost on artificially selecting valuable information from a mass of data, it is very meaningful to build automatic text summarization systems. Existing summarization algotihems can be roughly classified into two categories. *Extractive summarization* focuses on selecting key words, phrases or sentences from text and reproducing them as summary. *Abstractive summarization* aims to understand the source text entirely, capture its salient idea and generate a summary which may contain vocabularies unseen in the source text. In this paper, we focus on the latter, which acts in a similar way to humans.

The recent works based on deep learning, which cast abstractive summarization as mapping an input sequence into another output sequence, utilize sequence-to-sequence (seq2seq) models and have achieved state-of-the-art performance [1] [2]. However, seq2seq models commonly use maximum likelihood estimation (MLE) principle for training, which suffers from exposure bias in the inference stage [3]. Such a discrepancy between training and inference incurs accumulatively along with the sequence and will become prominent as the length of sequence increases. So, the coherence and readability of generated summaries are still not satisfactory, especially applying seq2seq model on long articles directly [4], which performs even worse than some traditional machine learning methods such as LexRank [5].

Motivated by this background, we adopt a thoroughly different training objective, targeting at directly minimizing the difference between generated summaries and the ground truth. To achieve this target, inspired by the recent success of GAN in machine translation (called ANMT) [6], we apply an adversarial training framework for text summarization. In this framework, besides the typical sequence generative model, a discriminator is introduced to distinguish automatically generated summary from the ground truth. The generator and discriminator are co-trained: the discriminator improves its performance by learning from more and more training samples; and the generator gets feedback from discriminator and improve itself such that it can successfully cheat the discriminator. To guarantee both modules are optimized, we use stochastic policy in reinforcement learning (RL) [7]. In this way, the generated summaries are teacher forced to be as close as possible to real ones.

2375-0197/18/$31.00 ©2018 IEEE
DOI 10.1109/ICTAI.2018.00045

IEEE computer society

Although both text summarization and machine translation are natural language processing tasks, there are some unique problems in summarization. Summarization can be cast as an optimal compression of the original document in a *lossy manner* [1] rather than one-to-one word-level alignment. So, summarization models should pay attention to potential primary parts of the source text. From another perspective, words in summary (the target text) have *global context dependency* with the source text, whereas words in translation merely have dependency with the source text in a certain range, such as intra-phrase or intra-sentence. So, the longer the source article, the more difficult to generate its summary. These differences motivates us to propose new network architectures: in the generator, we introduce the attention mechanism on both encoder and decoder; in the discriminator, we design a triple-RNNs model to capture the global contextual features of the source text and the summary independently.

The main contributions in this work are as follows: (i) We apply the sequence generative adversarial training protocol to text summarization, and show that it already outperforms state-of-the-art methods on both English and Chinese corpora. (ii) Motivated by the nature of summarization, we propose a novel discriminator model beyond the off-the-shelf one and show that it provide additional improvement in performance. (iii) The effectiveness of our model is more prominent in long text summarization, which demonstrates that our model also applies to long source text.

II. Related Work

Extractive Summarization Most summarization models in the past are extractive, generally utilizing machine learning techniques and graph-based methods to extract key words, phrases or sentences to be included in the summary. Traditional machine learning methods include Hidden Markov Models [8], decision tree, maximum entropy [9], support vector machines(SVM) and etc. Graph methods, which are influenced by PageRank algorithms such as TextRank [10] and LexRank [11], are more effective on long text summarization.

Abstractive Summarization Previous notable works include unsupervised topic detection method [12], phrase-table based machine translation approaches and quasi-synchronous grammar approaches [13]. With the successful application of deep learning models in NLP tasks, recent works have casted abstractive summarization as a sequence transformation problem and naturally adopted seq2seq models [14] [15]. A seq2seq framework consists of two neural networks (commonly CNN or RNN): one for encoding the input sequence into a fixed length vector C, and the other for decoding C and outputting the predicted sequence. Besides the basic neural network architecture, [16] [17] [18] append attention mechanism to the decoder allowing it to look back at parts of the encoded input sequence while the output is generated, and gain higher rouge score than traditional machine learning methods. In above seq2seq framework, the training strategy is Maximum Likelihood Estimation (MLE),i.e., to maximize the per-word likelihood of target summary conditioned on the source text. This strategy faces many challenges in generating sequences for discrete tokens, such as the difficulty of approximating intractable probabilistic computations.

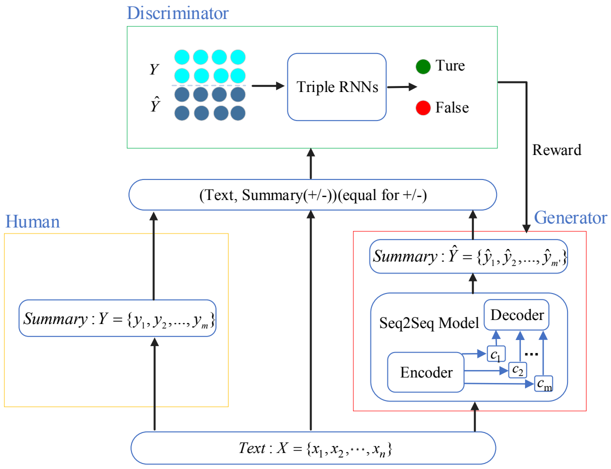
Discriminator
Y
$\hat{Y}$
Triple RNNs
Ture
False
Reward
(Text, Summary(+/-))(equal for +/-)
Human
Generator
$Summary: \hat{Y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_{m'}\}$
Seq2Seq Model
Decoder
Encoder
c_1
c_2
c_m
$Summary: Y = \{y_1, y_2, ..., y_m\}$
$Text: X = \{x_1, x_2, \cdots, x_n\}$

Fig. 1. Adversarial Text Summarization Framework

Application of GAN in NLP Ingeniously bypassing the difficulty of MLE learning, [19] proposed an alternative training methodology GAN, in which the training procedure is a *minmax* game between two models: a generator model to generate data, a discriminator model to distinguish generated data from real data. This framework has been proved tend to generate natural-looking samples in theory [20] and has gained remarkable successes in image generation [21]. However, there are problems in applying GAN in natural language generation tasks, such as the difficult gradient update and partially generated sequence evaluation. To solve these problems, [7] proposed SeqGAN which modeled the generator as a stochastic policy in reinforcement learning (RL) to generate discrete token sequences, and apply it in Chinese poem generation. Adopting similar training strategy, [6] applied GAN in Machine Translation named ANMT model, and showed that it significantly outperforms existing seq2seq models. Subsequently, [22] made a preliminary attempt in using GAN framework in summarization, which used almost the same model in ANMT.

Although, in the latest work, there are some attempts on adversarial training in natural language processing tasks, the efforts are very limited. Our work is among the first endeavors to explore the potential of acting in this way, especially for abstractive summarization. We design our model to partially solve the problems in text summarization, and use above notable methods including LexRank, Abs, ANMT as important baselines.

III. Adversarial Text Summarization Framework

The overall framework of our adversarial text summarization is shown in Figure 1. The input is a text sequence $X = \{x_1, x_2, \ldots, x_n\}$ consisting of n words, where x_i is

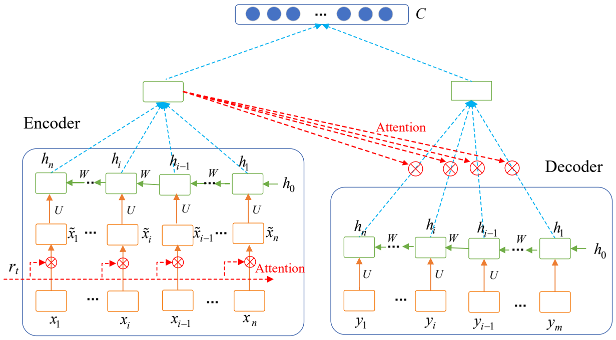

Fig. 2. The Seq2seq Generator Framework

the i-th word and $i < n$. Given the source text X, let $Y = \{y_1, y_2, \dots, y_m\}$ denote the ground-truth summary, and $Y' = \{y'_1, y'_2, \dots, y'_{m'}\}$ denote the automatically generated summary, where remarkably $m < n$, $m' < n$ and m' maybe not equal to m. We use *Generator* (shown in the red box) to transform the original text X into summary Y' under a typical sequence-to-sequence framework.

As mentioned above, our target is to force Y' to be as close as possible to Y. In order to achieve this goal, we introduce the adversary *Discriminator* (shown in the green box) to distinguish Y' from Y, which is based on triple recursive neural networks. The *Generator* and *Discriminator* are co-trained: we sample (X, Y) as positive instance and (X, Y') as negative one to train the *Discriminator*; simultaneously, we use strategy gradient to train *Generator* according to the reward from *Discriminator*.

A. Generator with Double-Attention

As shown in Figure 2, the *Generator*, which is called G for short, is an encoder-decoder model for sequence transformation. The goal of G is to generate a summary Y' for a given text x, which is formalized as $y'_i \sim G(Y'_{1:i-1}|X_{1:n})$, where $Y'_{1:i}$ means the generated partial summary at i-th step and $i \leq m'$.

In order to find useful part of the source text, we introduce the IARNN-WORD attention mechanism [23] on X, which solved the attention offset problem. Instead of using the words in original text to the encoder, we weight the words representation according to the encoder attention as follows:

$$\alpha_i = \sigma({r_t}^T M_{ik} x_i) \tag{1}$$

$$\tilde{x}_i = \alpha_i * x_i \tag{2}$$

where M_{ik} is an attention matrix which transforms the source text representation r_t into the word embedding space.

By this operation, the input source text are distilled by the above attention process. Then, we can fed the distilled whole source text $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n\}$ to the traditional encoder model. In this work, we adopt GRU instead of LSTM as building block for RNN both in encoder and decoder, because it has show advantages in many tasks.

Here, we redefine a conditional probability for G in the following:

$$G(y'_i|Y'_{1:i-1}, \tilde{X}) = g(y'_{i-1}, s_i, c_i) \tag{3}$$

Where s_i is the hidden status unit in the decoder, and c_i is the context vector at step i. For standard GRU decoder, the hidden status s_i is a function of the previous step status s_{i-1}, the previous step output y'_{i-1}, and the i-th context vector:

$$s_i = f(s_{i-1}, y'_{i-1}, c_i) \tag{4}$$

In order to generate the i-th word of the summary, G utilizes all input information of X, and the decoder get each word which have a unique Context Vector corresponding with it. The Context Vector is defined as follows:

$$c_i = \sum_{j=1}^{n} \beta_{ij} h_j \tag{5}$$

The weight β is defined as follows:

$$\beta_{ij} = \frac{exp(e_{ij})}{\Sigma_{k=1}^{n} exp(e_{ik})} \tag{6}$$

Where $e_{ij} = a(s_{ij}, h_i)$ is called the alignment model, which evaluates the matching degree of the j-th word of text and the i-th word of summary. This is a traditional attention mechanism used on the decoder.

B. Discriminator with Triple RNNs

The discriminator D is used to differentiate generated summary from real as much as possible. This is a typical problem of binary classification. Previous studies have shown that RNN is more suitable for text processing task, especially for long text, whether in feature selection or classification [24] [25]. So, we use RNN as the basic module in the discriminator architecture. Furthermore, considering text summarization a text compression problem rather than text alignment, we consider that rich feature of the original space (text space) are beneficial to D. Based on this intuition, we use two RNNs for capturing the contextual features of source (text) and target (summary) separately, and another one for classifying the given summary to “Generated” or “Real”. As shown in Figure 3, RNN_T and RNN_S are used for feature selection, while RNN_C are used for classification.

In order to prevent collapse mode, we use mini-batch method to train the discriminator. Here, we use human summary (X, Y) as positive instance while sampled summary (X, Y') as negative one, where $Y' \sim G(\cdot|X)$. At the beginning, we use the same batch size to randomly feed (X_i, Y_i) and (X_i, Y'_i) into the discriminator. For each text-summary pair fed to RNNs, $H_{content}$ is a hidden state vector of the source text and $H_{summary}$ is a hidden state vector of the summary. These two RNNs for feature extraction shared parameters.

Then, the abstract feature vector pairs $H = [(H^c_0, H^s_0), (H^c_1, H^s_1), \dots, (H^c_k, H^s_k)]$ are feed into the third RNN, which acts as a binary classifier. The final layer is a softmax layer which gives the probability that (X, Y)

is from ground-truth data, i.e. $D(X, Y)$. The optimization target of *D* is to minimize the cross-entropy loss for binary classification.

C. Policy Gradient for Training GAN

The adversarial summarization framework aims to encourage the generator to generate summaries that make the discriminator difficult to distinguish them from real ones. So, the ultimate goal function of training is:

$$\min_G \max_D V(D, G) = E_{(x,y)\sim P_r(X,Y)}[logD(X, Y)] + E_{(X,Y')\sim P_g(X,Y')}[1 - logD(X, Y')] \tag{7}$$

Where $P_r(X, Y)$ is text-summary pair sampled from human and $P_g(X, Y)$ is that from generator, $Y' \sim G(\cdot|X)$. That is, *G* tries to produce high quality summary to cheat *D*, meanwhile *D* tries to distinguish between the generated summary and the ground-truth.

If we use the above objective function to train *D* directly with sequence data, the gradient cannot be passed when the generator parameter is updated.The reason is that the output of generator is sampled discretely, so it is difficult to calculate the gradient for reverse propagation of gradient. In order to solve this problem, we introduce Reinforcement Learning (RL) mechanism. It is generally a markov decision process, performing an action y_i based on the state s_i with $Reward(s_i, y_i)$, where s_i denotes decoding result of the previous $i - 1$ words Y_{i-1}. A series of performed actions are called a "strategy" or "strategy path" θ^π. The target of RL is to find out the optimal strategy which can earn the biggest prize:

$$\theta^\pi_{best} = \arg\max_{\theta^\pi} \sum_{A_i \in \theta^\pi_{best}}^{i} Reward(s_i, y_i) \tag{8}$$

RL strategy can evaluate each possible action in any state through the environment feedback of reward, and find out one action to maximize the expected reward $E(\sum_{y_i \in \theta^\pi}^{i} Reward(s_i, y_i), \theta^\pi)$. Based on this, we assume that the generated summary is rewarded from the real summary by *D*, denoted as $R(X, Y')$. We denote parameters in the framework of encoder-decoder as θ, then our objective function is expressed as maximizing the expected reward of generated summary based on RL:

$$\begin{aligned} \theta^\pi_{best} &= \arg\max_\theta \mathbf{E}(R(X, Y')) \\ &= \arg\max_\theta \Sigma_X \Sigma_{Y'} P_\theta(X, Y')R(X, Y') \\ &= \arg\max_\theta \Sigma_X P(X) \Sigma_{Y'}(Y'|X)R(X, Y') \end{aligned} \tag{9}$$

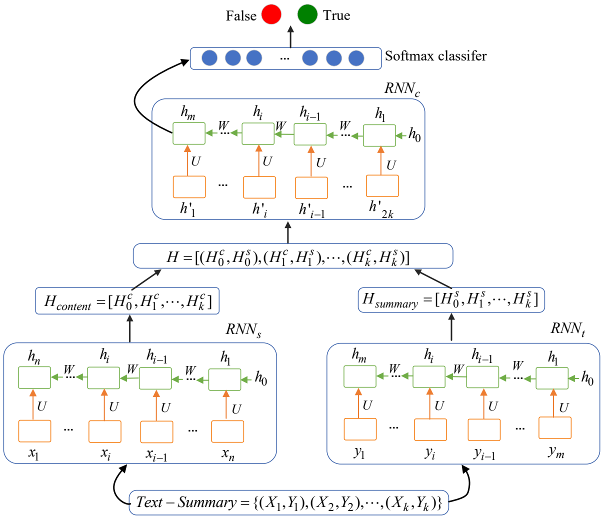
False
True
Softmax classifer
RNN_c
h_m
h_i
h_{i-1}
h_1
h_0
W
U
h'_1
h'_i
h'_{i-1}
h'_{2k}
$H = [(H_0^c, H_0^s), (H_1^c, H_1^s), \cdots, (H_k^c, H_k^s)]$
$H_{content} = [H_0^c, H_1^c, \cdots, H_k^c]$
$H_{summary} = [H_0^s, H_1^s, \cdots, H_k^s]$
RNN_s
h_n
x_1
x_i
x_{i-1}
x_n
RNN_t
y_1
y_i
y_{i-1}
y_m
$Text - Summary = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_k, Y_k)\}$

Fig. 3. The Triple-RNNs Discriminator Framework

Where $P_\theta(X, Y')$ denotes the probability of a text-summary pair (X, Y') under the parameter θ. We redefine he right-hand side of equation (9) as J_θ, which is the expectation of reward when *G* gets the optimal parameter. The probability distribution of each text-summary pair (X_i, Y'_i) can be regarded as a uniform distribution:

$$\begin{aligned} J_\theta &= \Sigma_X P(X) \Sigma_{Y'} P_\theta(Y'|X)R(X, Y') \\ &\approx \frac{1}{n} \Sigma_{i=1}^n R(X_i, Y'_i) \end{aligned} \tag{10}$$

Whose gradient w.r.t. is:

$$\begin{aligned} \nabla J_\theta &= \Sigma_X P(X) \Sigma_{Y'} R(X, Y') \nabla P_\theta(Y'|X) \\ &= \Sigma_X P(X) \Sigma_{Y'} R(X, Y') P_\theta(Y'|X) \frac{\nabla P_\theta(Y'|X)}{P_\theta(Y'|X)} \\ &= \Sigma_X P(X) \Sigma_{Y'} R(X, Y') P_\theta(Y'|X) \nabla \log P_\theta(Y'|X) \\ &\approx \frac{1}{n} \Sigma_{i=1}^n R(X_i, Y'_i) \nabla \log P_\theta(Y'_i|X_i) \end{aligned} \tag{11}$$

The gradient approximation is used to update θ , where α denotes learning-rate:

$$\theta^{i+1} = \theta^i + \alpha \nabla J_{\theta^i} \tag{12}$$

As a result, the key to gradient optimization is to calculate probability of generated summary. So, as the model parameter updates, our model will gradually improve the summary and reduce the loss. The expectation of the reward can be approximated by sampling methods. In the training process, the weakening of one side will lead to the interruption of the combat, i.e. the mode-collapse problem. So, we adopted Monte-carlo search to record the partial decoding and supplement its subsequent sequence, calculating the mean of all possible rewards. To be specific, when $t \neq n$, the decoding result is just a partial one whose reward is $R(X_i, Y'_{1:i})$.

Based on policy gradient, the adversarial *Discriminator* guide the generator to generate similar summaries as the ground-truth. In the ideal case, the distribution of generated summaries and that of real ones will overlap completely.

IV. Experiments and Results

A. Datasets and Evaluation

We report the experimental results on both short text summarization and long text summarization. We use CNN/Daily Mail corpus [1] [26] which is an English multi-sentence summarization dataset consists of online news articles. This corpus has 286,817 training pairs, 13,368 development pairs and 11,487 test pairs. It's worth noting that, most of comparative models merely use the first 2 sentences of a document (about 100 words) as the source text. So, we consider the actual corpus as a short text corpus. For long text summarization, we use the Chinese NLPCC corpus [2], consisting of training/development/test corpus with 50k, 2.5k and 2.5k text-summary pairs respectively. In the training set, the source document contains 789 words for longest while the summaries consist of 104 words.

We evaluate our models using the common assessment metric Rouge-score [5]. The basic idea of rouge score is to calculate the co-occurrence frequency of unit [27] between a candidate summary and a set of reference summaries. Our evaluation is based on three variants of Rouge, namely, Rouge-1(unigrams), Rouge-2(bigrams) and Rouge-L (longest-common substring LCS) are used.

B. Comparative Methods

We aim to verify the effectiveness of the adversarial framework, the triple RNNs discriminator, and double-attention mechanism respectively. So, we use some state-of-the-art seq2seq models and traditional machine learning methods as comparative methods.

- **LexRank:** As mentioned above, [11] outperforms existing deep learning models on long text summarization, and it is the best machine learning method on DUC2004 Task-2 and 4. So, we use LexRank as an important comparative method to evaluate our model on long text.
- **Abs:** [18] is the baseline attention-based seq2seq model which relies on LSTM network encoder and decoder. It uses first 2 sentences of a document with a limit at 120 words, 4-layers bidirectional encoding and 200k vocabulary.
- **Abs+INRNN:** We extend ABS by introducing attention on the encoder, referring to [23]. We compare this model to ABS, in order to verify the effectiveness of attention on encoder.
- **Abs+** [1] is an enhanced version of ABS, which relies on a range of separate extractive summarization features that are added as log-linear features in a secondary learning step with minimum error rate training. ABS+ achieves state-of-the-art performance among seq2seq models on CNN/Daily mail corpus. It uses the first 2 sentences of a document with a limit at 120 words 4-layers bidirectional en-coding and 150k vocabulary.
- **DeepRL:** [28] is a new training method that combines standard supervised word prediction and reinforcement learning (RL). It uses two 200-dimensional LSTMs for the bidirectional encoder and one 400-dimensional LSTM. The input vocabulary size is limited to 150k tokens.
- **ANMT:**The recent attempt of putting forward GAN framework in the neural machine translation [6] and text summarization [22], in which the discriminator is a specially designed CNN model.

C. Implementation Details

For the experiments on English dataset CNN/Daily mail, we set the dimensions of GRU hidden state and word embeddings to 128. Vocabulary size is set as 50k. The maximum length of text and summary is 350 and 50 respectively. For NLPCC corpus, we set the dimensions of GRU hidden state and word embeddings to 512. The maximum length of text and summary is 800 and 120 respectively. The encoder and decoder shared a 75k-size vocabulary.

For Generator, we used 2-layer GRUs both in encoder and decoder. We adopt Adam optimization algorithm [29], starting with an empirical learning-rate of 0.5. For Discriminator, Triple-RNNs use LSTM units and the learning rate is set as 0.2. The settings of hidden state layer and optimization algorithm in D and G are consistent. Before training the discriminator, we sampled the generated summary and the real summary randomly. Due to finiteness of generated summary, we use mini-batch strategy to feed text-summary pairs into the discriminator, in case of collapse mode. The minibatch is usually set as [32,64,128].

Our implementation is fully based on tensorflow1.4 [30]. We used two TITAN X (Pascal) GPU to train our model and comparative approaches. All models took about 4 hours per epoch on average. While most of comparative methods converged within 20 epochs, our model converged within 5 epochs based on pre-train, which demonstrated the lower computation cost.

In order to achieve the best results of comparative models, we used the parameter settings in the corresponding reference papers. In particular, [22] is not the original experimental setup, and we set 4-layer and 150k vocabulary.

D. Experimental Results

Table 1 and Table 2 show the results of our models on CNN/Daily mail corpus (short text summarization) and NLPCC corpus (long text summarization) respectively. Due to the limitation of hardware resources, we use a relatively small vocabulary. Even so, we achieved more promising results than other models.

From Table 1, we can observe the following results. Firstly, the baseline GAN model ANMT outperforms the ABS+ model with statistical significance, which demonstrates that the adversarial framework behaves better than seq2seq methods on the

[1] https://github.com/deepmind/rc-data

[2] http://tcci.ccf.org.cn/conference/2015/pages/page05_evadata.html

TABLE I
ROUGE-SCORE ON CNN/DAILY MAIL CORPUS

System	$Rouge_1$	$Rouge_2$	$Rouge_L$
LexRank	21.23	8.98	19.15
Abs	29.47	11.91	26.14
Abs+INRNN	33.15	13.20	26.36
Abs+	35.46	13.30	32.65
DeepRL	39.87	15.82	***36.90**
ANMT	39.92	17.65	36.71
ATRNNs	***41.56**	***18.42**	36.68

TABLE II
ROUGE-SCORE ON NLPCC CORPUS

System	$Rouge_1$	$Rouge_2$	$Rouge_L$
LexRank	22.03	0.72	13.01
Abs	11.35	1.67	14.17
Abs+INRNN	24.49	8.72	21.81
Abs+	26.82	9.19	24.12
DeepRL	29.90	10.36	26.59
ANMT	29.91	10.53	27.21
ATRNNs	***31.40**	***10.66**	***27.58**

TABLE III
EXAMPLES FROM THE CNN/DAILY MAIL TEST DATASET SHOWING THE OUTPUTS OF ABS AND ATRNNS MODELS, AFTER TOKENIZING, TRUNCATING TO 350 TOKENS AND REPLACING ARABIC NUMBERS IN "TAGNUM"

S(1):	two amish girls , apparently abducted (189 words)
R:	new : two girls found safe , authorities tell cnn
AB:	new : girl found TAGNUM girl say
AT:	new TAGNUM girl found safe TAGNUM girl say cnn
S(2):	paintings said gangster reggie kray(350 tokens)
R:	three paintings killer expected fetch TAGNUM
AB:	paintings killer killer fetch TAGNUM year
AT:	TAGNUM paintings killer killer expected fetch fetch TAGNUM TAGNUM
S(3):	britain launch world 's first spaceport (350 tokens)
R:	britain leading space race world 's first non - american spaceport
AB:	father spaceport daughter son daughter TAGNUM spaceport son TAGNUM kill first crash
AT:	uk uk early leading space big business uk TAGNUM spaceport space space space

text summarization task. The second, our model achieves the best performance on Rouge-1 and Rouge-2 score. In particular, it even surpasses the performances of other models with 4-layer RNNs and large vocabularies. This result demonstrates that the triple-RNNs discriminator performs better than the convolutional one. Besides, the performance of Abs+INRNN goes beyond Abs and is almost close to Abs+ (which introducing linguistic information), which proves that it is very effective to introduce the attention mechanism on encoder.

However, our Rouge-L score is not the highest one, which is likely due to two reasons: On one hand, in order to show that our model is also competitive in short text summarization, we truncated the length of summary to 50, which resulted in the shorter length of the longest-common substring (LCS). On the other hand, we set up a larger decoder vocabulary, which makes the words in generated summaries have more diversity. For example, the keywords in a human summary is "call" while its corresponding generated summary uses the synonym "phone". Although this setting results in the lower Rouge-L score, the generated summaries are not worse.

Table 2 reaches similar results as Table 1, and some results are notable. On NLPCC corpus, our ATRNNs model achieves the best summarization performance on all the Rouge score. Compared with the short essay, our model is more suitable for longer text. It is worth noting that LexRank get better Rouge-1 score than Abs, although its Rouge-2 score is the lowest. This result showed that directly applying the seq2seq model to text summarization performs worse than traditional machine learning methods, which is mentioned above.

E. Examples Analysis

In order to analyze the reasons of improving the performance, we compare the generated summary, Abs summary and human summary. The source texts(S), human summary(R), Abs summary(AB) and ATRNNs(AT) summary are shown in Table3. From these cases, we can find that our model ATRNNs are able to capture some implicit information and make semantic reasoning based on "understanding" the full text. For example, our result for S(2) is almost the same as human summary after removing redundant words. It's very interesting that, "three" in the original text is replaced with "TAGNUM" as the output, although we just preprocess the Arabic numerals in the corpus. In S(3), the result of Abs is completely inconsistent with human summary. However, our model still captures a lot of key information from the original text.

V. CONCLUSION

In this work, we propose a new adversarial training framework for long text summarization. In such a framework, we teach the generator to generate analogous human summary, which is achieved via introducing a discriminator which try it best to distinguish the generated summaries from the real ones. In order to handle long text better, we adopt the attention mechanism both for encoder and decoder in the generator, and design a Triple-RNNs discriminator. Our model got promising results in experiments.

There are several problems need to be resolved in the future work. Firstly, the generated summaries still consist of repeating phrases, so we will modify our generative model to make the decoder takes into account which words have already been generated. The second, our model is still a supervised learning one relying on high-quality training datasets which is scarce. So, we will study an unsupervised or semi-supervised framework which can be applied to the text summarization task.

References

[1] R. Nallapati, B. Zhou, C. dos Santos, Ç. glar Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *CoNLL 2016*, p. 280, 2016.

[2] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 93–98.

[3] S. Bengio, O. Vinyals, N. Jaitly *et al.*, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[4] S. Wang, X. Zhao, B. Li *et al.*, "Integrating extractive and abstractive models for long text summarization," in *Big Data (BigData Congress), 2017 IEEE International Congress on*. IEEE, 2017, pp. 305–312.

[5] S. Liu, "Cs585 project report long text summarization using neural networks and rule-based approach," 2017.

[6] L. Wu, Y. Xia, L. Zhao *et al.*, "Adversarial neural machine translation," *arXiv preprint arXiv:1704.06933*, 2017.

[7] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient." in *AAAI*, 2017, pp. 2852–2858.

[8] J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2001, pp. 406–407.

[9] L. Ferrier, "A maximum entropy approach to text summarization," *School of Artificial Intelligence, Division of Informatics, University of Edinburgh*, 2001.

[10] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.

[11] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[12] D. Zajic, B. Dorr, and R. Schwartz, "Bbn/umd at duc-2004: Topiary," in *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*, 2004, pp. 112–119.

[13] T. Cohn and M. Lapata, "Sentence compression beyond word deletion," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 137–144.

[14] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1700–1709.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[16] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," pp. 379–389, 2015.

[17] R. Nallapati, B. Xiang, and B. Zhou, "Sequence-to-sequence rnns for text summarization," 2016.

[18] P. J. Liu and X. Pan, "Text summarization with tensorflow," *Google Research Blog. Google Brain Team*, vol. 24, 2016.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[20] F. Huszár, "How (not) to train your generative model: Scheduled sampling, likelihood, adversary?" *stat*, vol. 1050, p. 16, 2015.

[21] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in neural information processing systems*, 2015, pp. 1486–1494.

[22] L. Liu, Y. Lu, M. Yang *et al.*, "Generative adversarial network for abstractive text summarization," *arXiv preprint arXiv:1711.09357*, 2017.

[23] B. Wang, K. Liu, and J. Zhao, "Inner attention based recurrent neural networks for answer selection," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 1288–1297.

[24] Y. Wu, D. Bamman, and S. Russell, "Adversarial training for relation extraction," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1778–1783.

[25] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[26] K. M. Hermann, T. Kocisky, E. Grefenstette *et al.*, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.

[27] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004.

[28] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *arXiv preprint arXiv:1705.04304*, 2017.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] M. Abadi, A. Agarwal, P. Barham *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.