2019 IEEE 13th International Conference on Semantic Computing (ICSC)

Generative Adversarial Network with Policy Gradient for Text Summarization

Banafsheh Rekabdar
Department of Computer Science
Southern Illinois University
Carbondale, IL, 62901
Email: brekabdar@cs.siu.edu

Christos Mousas
Computer Graphics Technology
Purdue University
West Lafayette, IN, 47907
Email: cmousas@purdue.edu

Bidyut Gupta
Department of Computer Science
Southern Illinois University
Carbondale, IL, 62901
Email: bidyut@cs.siu.edu

***Abstract*—Abstractive text summarization is the task of generating meaningful summary from a given document (short or long). This is a very challenging task for longer documents, since they suffer from repetitions (redundancy) when the given document is long and the generated summary should contain multi-sentences. In this paper we present an approach for applying generative adversarial networks in abstractive text summarization tasks with a novel time-decay attention mechanism. The data generator is modeled as a stochastic policy in reinforcement learning. The generator's goal is to generate summaries which are difficult to be discriminated from real summaries. The discriminator aims to estimate the probability that a summary came from the training data rather than the generator to guide the training of the generative model. This framework corresponds to a minimax two-player game. Qualitatively and quantitatively experimental results (human evaluations and ROUGE scores) show that our model can generate more relevant, less repetitive, grammatically correct, preferable by humans and is promising in solving the abstractive text summarization task.**

I. Introduction

Text summarization is the problem of generating a short, accurate, and fluent summary of a longer text document. Text summarization methods are either extractive or abstractive. Abstractive approaches are more challenging rather than extractive ones since a deeper analysis of the text is required. Abstractive methods in the area of sentence compression or revision [1] just focused on paraphrasing the text without taking in to account for an abstract representation for content selection. These techniques suffer from inaccurate encoding of long documents, and generating unnatural redundant output, especially when generating multi-sentences summary. In such problems attention mechanism is necessarily to prevent looking at the sames parts of the input for decoding each generated summary word.

In this paper, we propose an abstarctive text summarization method based on generative adversarial network (GAN) to address the limitations mentioned above to generate summaries which are hard to discriminate from the ones written by the human. The motivation is to generate a readable, natural summary from an input article utilizing a discriminator network which automatically learns how human written summaries look like. Following this motivation, we build a generative adversarial network where we train two adversarial models: a generator that generates the summary based on the input article; and a discriminator network, conditioned on the input article to predict the probability that the generated summary is real (human generated). In the training step, the generator's goal is to generate a summary looking like the one written by human to fool the discriminator and makes it to believe that it is not machine generated. On the other hand the discriminator's goal is to improve it's ability to distinguish the machine generated summary from the real one and to provide feedback to the generator.

GANs are proved to be very powerful in computer vision and image generation areas since they were originally designed to generate real-value (differentiable) data. Thus generating discrete outputs (sequences of discrete tokens) is a challenging task for them. This is due to the discrete nature of text making it infeasible to propagate the gradient from the discriminator back to the generator as in standard GAN training [2]. Following other researchers to solve this problem in the domain of text generation [3] [2], translation [4], and summarization [5] [6], we use Reinforcement Learning (RL) to train the generator.

In summary the key contributions of this paper are as follows: 1) To the best of our knowledge, this work is among the first few endeavors to use the generative adversarial networks in abstractive text summarization tasks, 2) We introduced a time-decay attention mechanism to address the repeating phrase problem, 3) The generator is modeled in a reinforcement learning setting.

II. Previous Work

Summarization is the task of generating a summary that contains the core information from the original text. Summarization methods fall in to two prominent categories: extraction-based [7] and abstraction-based [8]. Human written summaries are mostly abstractive [8]. Attentional encoder-decoder models for machine translation are very successful in the area of abstractive text summarization [9][10]. However these models are utilized to generate a summary for a short input article. In [11], the generated summaries are not more than 75 characters on the DUC-2004 dataset. Romain Paulus et al. introduced an encoder decoder model with a novel intra attention that attends over the input and continuously generated output separately, and a new training method that combines

978-1-5386-6783-5/19/$31.00 ©2019 IEEE
DOI 10.1109/ICSC.2019.00047

reinforcement learning (RL) with standard supervised word prediction methods to summarize long input text [8].

GANs are proved to be very powerful in computer vision and image generation areas since they originally designed to generate real-value (differentiable) data. Thus generating discrete outputs (sequences of discrete tokens) is a challenging task for them. This is due to the discrete nature of text making it infeasible to propagate the gradient from the discriminator back to the generator as in standard GAN training [2]. Researchers overcame this problem by using Reinforcement Learning (RL) to train the generator [3] [2]. Authors in [5] proposed an adversarial process for abstractive text summarization using policy gradient. Their generator and discriminator is based on bidirectional-LSTM and CNN respectively. Hao Xu et. al. in [6] proposed a novel adversarial reinforcement learning architecture for Chinese text summarization. They employed character-level model and append Text-Attention proposed by [9] in the generator. The architecture of their generator and discriminator is encoder-decoder LSTM and recurrent neural network (RNN) respectively.

In this paper we propose an approach for applying generative adversarial networks in abstractive text summarization tasks. Our summary generator is modeled as a stochastic policy in reinforcement learning. We use a novel time-decay attention mechanism in both encoder and decoder part of our generator .

III. Proposed Approach

In this section, we present the architecture of our proposed model in details. We interpret the summary generation process as a reinforcement learning setting in which a sequence of actions are taken (generating the tokens of the summary) based on a policy regulated by the generator in a generative adversarial network. In our work we employed the policy gradient method following the seqGAN [3]. Our model has 2 main parts, generator and discriminator. The generator aims to generate a summary indistinguishable from the human summary, based on the given source article. The discriminator, conditioned on the source article, tries to distinguish the machine-generated summary from the human written one.

A. Generator Model

The generator regulates the policy G_θ that generates the summary $Y_{1:T} = (y_1, ..., y_t, ..., y_T)$ given the source article $(X = x_1, ..., x_n)$ where Y is the vocabulary of candidate tokens. G_θ is a stochastic policy. We define this policy as follows: $G_\theta(y_t|Y_{1:t-1}, X)$. In each timestep t, it gives us the probability of generating y_t, given it's current state (the current produced tokens $(y_1, ..., y_{t-1})$ and the source article X).

1) LSTM encoder-decoder with attention mechanism: Our generator is a LSTM encoder-decoder as in [8] [9]. The encoder consists of a bidirectional LSTM, while the decoder consists of a uni-directional LSTM with attention mechanisms over the encoder-hidden states h_i^e and decoder hidden states h_t^d. h_i^e is defined in equation 1 as the concatenation of the encoder's backward and forward hidden states. The encoder reads the input sequence X. As in standard language-modeling, the decoder generate the tokens of the summary $Y_{1:T} = (y_1, ..., y_t, ..., y_T)$ auto-regressively, however, it is now conditioned on both the source input as well as what it has been generated to that point. In order to increase the readability of the summary and reduce the amount of repetitions, at each decoding time step t, we use a time-decay intra-temporal attention mechanism. This mechanism records previous attention weights over the encoded input tokens. This attention function helps to focus over different parts of the encoded input sequence at different encoding time steps [12] [9].

$$h_j^e = [h_j^{\rightarrow}][h_j^{\leftarrow}] \tag{1}$$

The attention score e_{ti} for the hidden input state h_j^e at decoding time step t is defined in equation 2. In this equation f is a bilinear function as in [8].

$$e_{tj} = f(h_t^d . h_j^e) \tag{2}$$

The following time-decay temporal attention function b_{tj}^e is used to penalize words that obtained high attentional scores in the previous time steps but in a decaying manner. It is intuitive that recent utterances should be penalized more than the least recent ones since the recent ones should not be repeated soon.

$$b_{tj}^{he} = \sum_{k=1}^{t-1} exp(e_{kj}.1/(t-k)) \tag{3}$$

$$b_{tj}^e = exp(e_{tj})/b_{tj}^{he} \tag{4}$$

We normalize the attention score b_{tj}^e over the input tokens to compute the input context vector c_t^e (equations 5 and 6).

$$a_{tj}^e = b_{tj}^e / \sum_{i=1}^{n} b_{ti}^e \tag{5}$$

$$c_t^e = \begin{cases} \sum_{j=1}^{n} a_{tj}.h_j^{\rightarrow} \\ \sum_{j=1}^{n} a_{tj}.h_j^{\leftarrow} \end{cases} \tag{6}$$

Apart from the intra-temporal attention mechanism on the encoded input sequence, we define a time-decay attention mechanism on the decoder part which considers previously generated tokens of the summary by the decoder. In this way we reduce the chance of generating the repeated tokens and phrases via the decoder. The attention score $d_{tt'}$ at decoding time step t for previously generated tokens of the summary (at previous decoding time steps t') is defined in equation 7. In this equation f is a bilinear function [8].

$$d_{tt'} = f(h_t^d . h_{t'}^d) \tag{7}$$

The following time-decay attention function $b_{tt'}^{hd}$ is used to penalize the weights of the previously generated words via decoder in a decaying manner. It's a time-aware attention mechanism that penalize the recent generated words more than the least recent ones by a factor of $1/(t-t')$.

$$b^{hd}_{tt'} = \sum_{k=1}^{t-1} exp(d_{tk}.1/(t-k)) \quad (8)$$

$$a^{d}_{tt'} = exp(d_{tt'})/b^{hd}_{tt'} \quad (9)$$

At each decoding time step t, the model computes a new decoder context vector c^d_t.

$$c^d_t = \sum_{j=1}^{t-1} a^d_{tj}.h^d_j \quad (10)$$

B. Discriminator Model

Following [13] we train a discriminator model $D_\phi(Y_{1:T}|X)$ to give us the probability that how likely a given summary is real. The D_ϕ is trained using positive and negative examples. The positive samples are pairs of input article and the human generated summary from the dataset. The negative samples are the pairs of the input article and the generated summary by G_θ or the input article and a random selected summary from the dataset. The discriminator provides expected end reward for the generator. Hence the generator model is updated by employing a policy gradient and Monto Carlo search on the basis of the discriminator's expected end reward. Since Convolutional neural network [6] are proved to be successful in the text classification and language processing areas, our discriminator is based on Convolutional neural network architecture to process the pairs of long input article and the summary to find out if the given summary is real or not.

C. Policy Gradient training

Following [3][4], the objective of the generator is to generate a summary to maximize the cumulative total reward. Formally it is defined as follows:

$$E_{G_\theta}[R] = \sum_{Y_{1:T}} G_\theta(Y_{1:T}|X).Q^{G_\theta}_{D_\phi}(Y_{1:T-1}, X, y_T) \quad (11)$$

where θ is the parameters in the generator in which we optimize it by performing gradient ascent on $E_{G_\theta}[R]$. $Y_{1:T}$ is the generated summary and X is the source article. $Q^{G_\theta}_{D_\phi}$ is the action-value function where the state is the generated summary so far; $Y_{1:T-1}$; given the source article X; and the action is generating y_T following the policy G_θ. The discriminator provides the probability that the generated summary is real. We consider this probability as the reward for the generator.

$$Q^{G_\theta}_{D_\phi}(Y_{1:T-1}, X, y_T) = D(Y_{1:T}, X) - b(Y_{1:T}, X) \quad (12)$$

In the above equation the $b(Y_{1:T}, X)$ is the baseline value to reduce the variance of the reward. Following [4] we set the baseline value as 0.5 (a constant) for simplicity. Based on [3][4] the discriminator is able to produce a reward signal only for a complete summary. Hence to estimate the action-value for an intermediate state (incomplete summary) $(Y_{1:t}, X)$, we use N-time Monte Carlo search under the policy of G_θ to sample unknown tokens of the summary $(Y_{1:T_i-t})$.

$$Y^1_{1:T_1}, ..., Y^N_{1:T_N} = MC^{G_\theta}((Y_{1:t}, X), N) \quad (13)$$

where T_i is the length of the summary sampled by the i'th Monte Carlo search and corresponds to the length of the real summary of the source article in the dataset. Thus the discriminator generates N rewards for one intermediate state (based on N-time Monte Carlo search), the final reward of the intermediate state is the average of those N rewards.

We re-train the discriminator after we obtained more realistic summaries from the generator. Following traditional GANs the objective function of the discriminor is as follows:

$$min - E_{X,Y \in P_{data}}[logD(X,Y)] - E_{X,Y \in G_\theta}[log(1 - D(X,Y))] \quad (14)$$

Each time after re-training the discriminator, we train the generator via optimizing it's parameters θ, by performing gradient ascent on $E_{G_\theta}[R]$ as follows:

$$\nabla E_{G_\theta}[R] = 1/T \sum_{t=1}^{T} E_{y_t \in G_\theta} \nabla log G_\theta(y_t|Y_{1:t-1}, X). Q^{G_\theta}_{D_\phi}(Y_{1:t-1}, X, y_t) \quad (15)$$

D. Training strategies

We first pre-train the generator with the maximum likelihood estimation. Next, we pre-train the discriminator with the combination of the real summary and the generated summary until the classification accuracy becomes α. Following [4] we make the discriminator to assign a reward of 1 to the real summaries during training the generator, which makes the generator more stable. We use two 200-dimensional LSTMs for the bidirectional encoder and one 400-dimensional LSTM for the decoder. The size of the input and output vocabulary size are 150,000 and 50,000 respectively. We used Adam with a batch size of 100 and a learning rate of 0.002. We set 30 for N in the Monto Carlo search, and 0.7 for the α in the discriminator. Input word embeddings are also 100-dimensional. The convolutional neural network consists of three pairs of convolution, max pooling layers, one Multilayer perceptron layer and one softmax layer, with $3*3$, $5*5$, $9*9$ convolution window sizes, $2*2$ pooling window size, 30 feature map size and 50 MLP hidden layer size.

IV. Experimental Results

We tested our approach on the CNN/Daily Mail dataset [14]. It consists of news stories in CNN and Daily Mail websites paired with multi-sentence human generated abstractive summaries. We followed the same pre-processing as explained in [9]. The dataset contains 287,113 training pairs, 13,368 validation pairs and 11,490 testing pairs. Following [8] we limited the size of the input and the output tokens to 800 and 100 respectively.

We use the full length F1 variant of Rouge scores (ROUGE-1, ROUGE-2, and ROUGE-L) to quantitatively evaluate our method. However, the most accurate way of evaluating generative models is the human judgment. Hence we conduct human evaluation of linguistic quality of the samples from a baseline and our method. Three different dimensions are assessed: grammatically, relevance (how relevant is the summary to the article and how good it digests the article), and readability (how meaningful and understandable the summary is). For human judgment, we randomly chose 30 samples from the dataset. For each sample we show the original article, the generated summary by a different model and the original summary to 5 human judges. They score them in the range of 1 (the worst) to 10 (the best). We obtain the total score for each dimension via averaging the scores over all the evaluators and the samples.

Table I and Table II are shown the results based on ROUGH scores and human judgment respectively. Based on Table I our approach performs better than all other methods in-terms of ROUGH-L score. As the results present, the time-decay attention mechanism improves the ROUGH-1, ROUGH-2, and ROUGH-L scores of our proposed model. The results of the human evaluation are shown in Table II. Based on this table, our approach obtained higher scores in terms of Grammatically and Relevance compared with the other model.

TABLE I
COMPARISON OF VARIOUS MODELS ON THE CNN/DAILY MAIL DATASET USING FULL-LENGTH F1 VARIANTS OF ROUGE. (+: WITH; -: WITHOUT)

	ROUGE-1	ROUGE-2	ROUGE-L
Lead-3 ([7])	39.2	15.7	35.5
SummaRuNNer ([7])	39.6	16.2	35.3
words-lvt2k-temp-att ([9])	35.46	13.30	32.65
RL +intra-attention ([8])	**41.16**	15.75	39.08
ML+RL + intra-attention ([8])	39.87	15.82	36.90
GAN ([5])	39.92	**17.65**	36.71
Proposed method + attention	37.87	15.71	**39.20**
Proposed method - attention	34.87	14.71	33.20

TABLE II
COMPARISON OF HUMAN JUDGMENT. (+: WITH; -: WITHOUT)

	Grammatically	Relevance	Readability
ML+RL + intra-attention ([8])	7.1	7.71	**7.21**
Proposed method + attention	**7.5**	**7.9**	6.1

V. CONCLUSION

Abstractive text summarization is a challenging task in particular when the given document is long and the generated summary should contain multi-sentences. In this paper we propose to use generative adversarial network in abstractive text summarization tasks with a novel time-decay attention mechanism. We evaluate the summaries generated by our proposed architecture both qualitatively and quantitatively and found that the generated summaries by our proposed model are more relevant, less repetitive, grammatically correct and preferable by human.

REFERENCES

[1] B. B. Klebanov, K. Knight, and D. Marcu, "Text simplification for information-seeking applications," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2004, pp. 735–747.

[2] W. Fedus, I. Goodfellow, and A. M. Dai, "Maskgan: Better text generation via filling in the _," *arXiv preprint arXiv:1801.07736*, 2018.

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

[4] Z. Yang, W. Chen, F. Wang, and B. Xu, "Improving neural machine translation with conditional sequence generative adversarial nets," *arXiv preprint arXiv:1703.04887*, 2017.

[5] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative adversarial network for abstractive text summarization," *arXiv preprint arXiv:1711.09357*, 2017.

[6] H. Xu, Y. Cao, Y. Shang, Y. Liu, J. Tan, and L. Guo, "Adversarial reinforcement learning for chinese text summarization," in *International Conference on Computational Science*. Springer, 2018, pp. 519–532.

[7] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents." in *AAAI*, 2017, pp. 3075–3081.

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

[9] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv preprint arXiv:1602.06023*, 2016.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[11] W. Zeng, W. Luo, S. Fidler, and R. Urtasun, "Efficient summarization with read-again and copy mechanism," *arXiv preprint arXiv:1611.03382*, 2016.

[12] B. Sankaran, H. Mi, Y. Al-Onaizan, and A. Ittycheriah, "Temporal attention model for neural machine translation," *arXiv preprint arXiv:1608.02927*, 2016.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[14] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.