Semantic-Preserving Abstractive Text Summarization with Siamese Generative Adversarial Net

Xin Sheng[1], **Linli Xu**[2,3]*, **Yinlong Xu**[1], **Deqiang Jiang**[4], **Bo Ren**[4]
[1] School of Computer Science and Technology, University of Science and Technology of China.
[2] Anhui Province Key Laboratory of Big Data Analysis and Application,
School of Computer Science and Technology, University of Science and Technology of China.
[3] IFLYTEK Co., Ltd. [4] Tencent Youtu Lab.
xins@mail.ustc.edu.cn, {linlixu,ylxu}@ustc.edu.cn,
{dqiangjiang,timren}@tencent.com

Abstract

We propose a novel siamese generative adversarial net for abstractive text summarization (SSPGAN), which can preserve the main semantics of the source text. Different from previous generative adversarial net based methods, SSPGAN is equipped with a siamese semantic-preserving discriminator, which can not only be trained to discriminate the machine-generated summaries from the human-summarized ones, but also ensure the semantic consistency between the source text and target summary. As a consequence of the min-max game between the generator and the siamese semantic-preserving discriminator, the generator can generate a summary that conveys the key content of the source text more accurately. Extensive experiments on several text summarization benchmarks in different languages demonstrate the effectiveness of the proposed method.

1 Introduction

Abstractive text summarization endeavors to produce a concise and fluent summary for a given text, while maintaining the key content and overall meaning. Previous attempts tackle this problem with either rule-based or statistical-based methods. Recently, with the successes obtained on the machine translation task (Sutskever et al., 2014; Sheng et al., 2020), the neural network based sequence-to-sequence framework is also applied to the abstractive text summarization task. Specifically, the sequence-to-sequence architecture consists of an encoder responsible for transforming the source sequence $\mathbf{x} = \{x_1, x_2, \ldots, x_{Tx}\}$ into an intermediate representation, and a decoder to generate a target sequence $\mathbf{y} = \{y_1, y_2, \ldots, y_{Ty}\}$ using the previously generated intermediate representation. Furthermore, to dynamically generate a context vector for a target word being generated, the attention mechanism (Bahdanau et al., 2014; Luong et al., 2015) is proposed to strengthen the sequence-to-sequence models, which enables the model to focus on the relevant parts of the source-side sequence. Based on the encoder-decoder framework, many variants of model structures, such as convolutional neural network (CNN) and recurrent neural network (RNN) are proposed (Bahdanau et al., 2014; Gehring et al., 2017). With the emergence of Transformer (Vaswani et al., 2017), which is based entirely on the attention mechanism, state-of-the-art performance is achieved on many sequence-to-sequence tasks. Nevertheless, for the task of abstractive text summarization, one of the dominant challenges is to maintain saliency, which requires the generated summary to convey the important information accurately. As shown in Figure 1, the key content of the source text "Chengdu become the 'Silicon Valley' in the west of our country" is

Source: 成都市软件和信息技术服务业近些年来一直保持快速增长势头，稳居中西部城市之首，**已成为我国西部“硅谷”**。《2013年度成都市软件和信息技术服务产业发展报告》日前发布……详情请见:@成都日报@成都发布
The software and information technology service industry in Chengdu has maintained the momentum of rapid development in recent years, ranking first among the cities in the central and western regions, **and has become the "Silicon Valley" in the west of our country**. "The 2013 Chengdu Software and Information Technology Service Industry Development Report" was released a few days ago... For details, please see: @Chengdu Daily@Chengdu post
Reference: 成都倾力**打造西部“硅谷”**
Chengdu strives to **build the Western "Silicon Valley"**
Generated: 成都发布软件和信息技术服务产业发展报告
Chengdu releases software and information technology service industry development report

Figure 1: The case of lacking saliency in abstractive text summarization. Bold text represents the key content, while the underlined parts represent the unimportant content.

*Corresponding author

Findings of the Association for Computational Linguistics: NAACL 2022, pages 2121 - 2132

accurately summarized in the reference, while the generated summary expresses the unimportant content "Chengdu releases software and information technology service industry development report".

Intuitively, the lack of saliency in summarization is usually caused by attending to wrong parts of the source text, inspiring many attention optimization methods for more accurate attention mechanism. Among them, (Lin et al., 2018) proposes a global encoding framework, which controls the attention information flow from the encoder to the decoder based on the global information of the source context. (Gui et al., 2019) proposes an effective method to regularize the attention weights from both global and local aspects. (Duan et al., 2019) introduces a novel attention mechanism, where the attention weights on relevant parts of the source side are encouraged while the attention weights on less relevant or irrelevant parts are discouraged with a softmax and a softmin function respectively. However, for these methods, the underlying nature of saliency, which is actually the sentence-level semantic consistency between the source text and the generated summary, is generally overlooked.

To explicitly maintain the semantic consistency, we propose a novel Siamese Semantic-Preserving Generative Adversarial Net (SSPGAN) for abstractive text summarization. In SSPGAN, different from conventional adversarial training (Goodfellow et al., 2014) which mainly focuses on how to generate more realistic data, a novel training paradigm is introduced to generate a summary that is more semantically consistent with the source text. Specifically, the proposed model consists of two adversarial modules which play a min-max game:

- A conventional neural encoder-decoder based generator, which aims to generate the summary sequence based on the input text.

- A siamese semantic-preserving discriminator. Different from the conventional discriminator in a generative adversarial net (GAN), in addition to distinguishing the real summary from the generated summary, it is also required to capture the semantic consistency between the source text and the target summary. And we adopt a pseudo siamese net to achieve that. Specifically, we aim to maximize the semantic similarity for a real sentence pair (text, real summary), while minimizing it for a generated sentence pair (text, generated summary).

During the training process, in terms of the authenticity and semantic consistency with the input source text, the generator aims to fool the discriminator into believing that its output is a human-generated summary, and the discriminator makes efforts not to be fooled by improving its ability to distinguish the machine-generated summary from the human-generated one. This kind of adversarial training achieves a win-win situation when the generator and the discriminator reach a Nash Equilibrium (Zhao et al., 2016; Arora et al., 2017; Guimaraes et al., 2017).

Different from conventional GANs, which assume the existence of a generator in a continuous space, in our proposed framework, the text summarization model is in fact not a typical generative model, but instead a probabilistic transformation that maps a source text to a target summary, both in a discrete space. To this end, we turn to a policy gradient method named REINFORCE (Williams, 1992), which can guarantee that both the two sub models are effectively optimized in an adversarial manner. In addition to the conventional reward, which is the estimated probability of the generated summary being discriminated as the real one, we also adopt the semantic similarity between the source text and the generated summary as a supplementary reward signal. Besides, we employ Transformer (Vaswani et al., 2017) as the basis of our discriminator to capture both the global and local features of the sentence.

The contributions of this work are three-fold:

- We propose a siamese net based discriminator to ensure the semantic consistency between the generated summary and the source text.

- A generative adversarial net based entirely on Transformer is proposed. As far as we know, this work is the first attempt to apply such framework into the text summarization task.

- Experimental results on both English and Chinese text summarization datasets show that the proposed model outperforms conventional GAN-based methods. And we also demonstrate that the proposed method can maintain semantic consistency from multiple perspectives.

2 Related Work

Automatic text summarization can be broadly divided into extractive and abstractive summarization.

The extractive methods simply extract important parts of the source text and reorganize them in a certain order (Jing and McKeown, 2000; Knight and Marcu, 2000; Neto et al., 2002). In comparison, abstractive text summarization is closer in principle to the process of manual summarization, which extracts the essential information of the source text and describes it in a shorter version as the abstractive summary. In this paper, we focus on abstractive text summarization.

Previous works on abstractive text summarization are mainly designed with statistical methods and rule-based methods (Banko et al., 2000; Dorr et al., 2003; Zajic et al., 2004; Cohn and Lapata, 2008). Recently, the sequence-to-sequence neural framework becomes predominant on the task of abstractive text summarization (Chopra et al., 2016; Nallapati et al., 2016; Li et al., 2017b). Later on, with the advent of Transformer (Vaswani et al., 2017), more and more works choose it as the base model in their frameworks.

For the abstractive text summarization task, out-of-vocabulary (OOV), repetitions and lack of saliency are three dominant challenges. To tackle the problem of OOV, some works introduce the pointer network and copy mechanism (Nallapati et al., 2016; See et al., 2017; Gu et al., 2016; Paulus et al., 2017). On the issue of repetitions, (See et al., 2017) adopts a coverage mechanism, which is inspired by the coverage vector from neural machine translation (Tu et al., 2016). Regarding saliency, some works (Duan et al., 2019; Gui et al., 2019) focus on how to optimize the attention mechanism, while (Zhu et al., 2021) tries to enhance the factual consistency with a fact corrector. Meanwhile, (Narayan et al., 2021) adopts the content planning to improve the performance of abstractive summarization model. However, the essence of saliency, which is the sentence-level semantic consistency between the source text and the generated summary, is intuitive yet usually overlooked.

The proposed training principle is based on adversarial learning (Goodfellow et al., 2014). In conventional adversarial training, a generator and a discriminator compete with each other, forcing the generator to produce high quality samples that can fool the discriminator. Adversarial training typically excels in image generation (Goodfellow et al., 2014), with less applications in natural language processing tasks (Yu et al., 2017; Li et al., 2017a), mainly due to the difficulty of propagating the signals from the discriminator to the generator through the discretely generated tokens. (Yu et al., 2017) addresses this issue with a reinforcement learning approach for sequence generation. Thus, the adversarial training paradigm can improve the model on the sentence-level instead of the vanilla token-level (e.g., maximum likelihood estimation).

To address the semantic inconsistency problem mentioned above, we introduce the paradigm of siamese net into GANs. Siamese net is a class of neural network architectures that contain more than one identical or different sub networks, which depends on whether the inputs are similar or not. Siamese net is generally used to measure the similarity between the inputs by comparing their corresponding output feature vectors, and can be broadly divided into two types: true siamese net and pseudo siamese net. The true siamese net contains identical sub networks which share the same architecture and network parameters, while the pseudo siamese net contains sub networks which have different parameters and even different architectures. Among the existing works, (Kenter et al., 2016) is the first to adopt siamese net into unsupervised sentence embedding learning. (Mueller and Thyagarajan, 2016) proposes MaLSTM to learn sentence similarities with Manhattan distance. (Neculoiu et al., 2016) considers similarity matching of a sentence pair as a binary classification task and replaces the Manhattan distance with cosine similarity. Recently, (Reimers and Gurevych, 2019) introduces the principle of siamese net to fine-tune BERT (Devlin et al., 2019) for better sentence embedding.

Different from previous GAN-based abstractive text summarization model in the work of (Liu et al., 2018), by incorporating siamese net into GANs, the generator can generate summaries which are more semantically consistent with the source texts. As far as we know, this work is the first attempt to apply siamese net to the GAN-based sequence-to-sequence generation task.

3 Siamese Semantic-Preserving Generative Adversarial Net

In this section, we introduce the architecture of the proposed Siamese Semantic-Preserving Generative Adversarial Net (SSPGAN) in detail. The model consists of two main components. The first component is a standard Transformer-based summary generator G (Figure 2). During adversarial training, the generator G is treated as an agent tak-

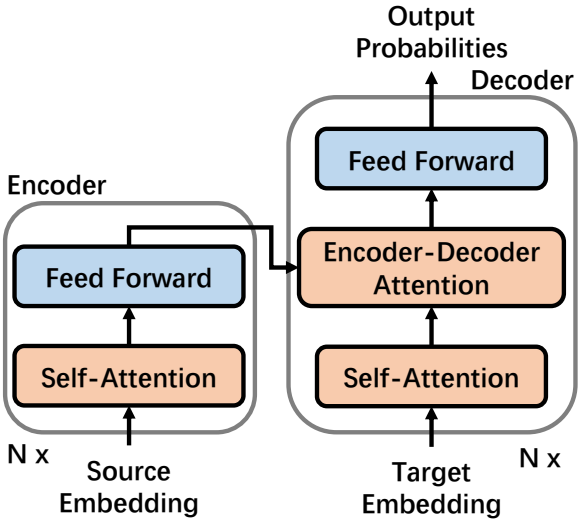

Figure 2: The summary generator, taking a conventional Transformer based encoder-decoder architecture (Vaswani et al., 2017), where the predicted word from the previous step serves as the input of the current step during inference. We omit some layers for brevity.

ing sequential actions (i.e., generating words) and trained using policy gradient given the reward of each generated word. The second component is a siamese network based discriminator D, which is also implemented based on the Transformer. On the one hand, the discriminator D is required to distinguish the generated summary from the real one. On the other hand, it aims to capture the semantic similarity between the source text and the target summary. Specifically, it is expected to maximize the semantic similarity for the real sentence pair (text, real summary), while minimizing the semantic similarity of the generated pair (text, generated summary). From these two perspectives, we compute a composite reward for each generated summary. Both the generator G and the discriminator D are iteratively trained. Figure 3 shows the overview of the adversarial training framework. In the following, we describe the generator G and the siamese semantic-preserving discriminator D in detail.

3.1 Generator

At time step t, the generator G takes an action (i.e., a word y_t) according to a stochastic policy $\pi_\theta(y_t|\mathbf{x}, \mathbf{y}_{t-1})$, where $\mathbf{x}$ is the input source text, $\mathbf{y}_{t-1} = [y_1, \ldots, y_{t-1}]$ is the previously generated partial summary, and θ is the parameter of the policy. We utilize the conventional Transformer based encoder-decoder framework (Vaswani et al., 2017) as the model of the policy. By sequentially generating each word y_t using the policy $\pi_\theta(.)$ until the end, a complete sentence $\mathbf{y}$ is generated. In conventional sequence-to-sequence learning, the model is trained to minimize the cross-entropy loss:

$$\mathcal{J}(\theta) = -\sum_{n=1}^{N}\sum_{t=1}^{T_n} \log \pi_\theta(\hat{y}_t^n|\mathbf{x}^n, \hat{\mathbf{y}}_{t-1}^n) \quad (1)$$

where N is the number of text-summary pairs, T_n is the length of the ground-truth summary $\hat{\mathbf{y}}^n$, Loss is the cross-entropy loss, $\hat{\mathbf{y}}_{t-1}^n$ and $\hat{y}_t^n$ are the ground-truth partial summary and word, respectively. Nevertheless, in adversarial training, these is no explicit supervised information for computing the cross-entropy loss. Hence, we adopt our discriminator D to assess the quality of the generated complete summary $\mathbf{y}^n$. Specifically, the discriminator D is responsible for calculating a reward using the generated summary $\mathbf{y}^n$ and the source text $\mathbf{x}^n$ (See Section 3.3 for details).

3.2 Siamese Semantic-Preserving Discriminator

Our discriminator D aims to not only distinguish the real summary from the generated one, but also capture the semantic similarity between the source text and the target summary. Here, the discriminator D is implemented based on the Transformer, as Transformer is capable of capturing both local and global sentence features. In the meantime, to capture the semantic similarity, the whole framework of the discriminator D is designed based on the siamese net (right panel in Figure 3).

Given the source text $\mathbf{x} = \{x_1, x_2, \ldots, x_{Tx}\}$ and the target summary $\mathbf{y} = \{y_1, y_2, \ldots, y_{Ty}\}$ (here $\mathbf{y}$ represents both the real and generated summary for simplicity), where x_t and y_t are the t-th words in the corresponding sequences. For the source text sequence $\mathbf{x}$, we take it as input of the Transformer encoder. After the processing of Transformer blocks, a hidden state sequence $h_\mathbf{x}$ will be produced:

$$h_\mathbf{x} = \{h_{x_1}, h_{x_2}, \ldots, h_{x_{T_x}}\} \quad (2)$$

where h_{x_t} is the hidden state corresponding to x_t in the input sequence. Thus, h_{x_t} contains not only the positional information, but also the global and local correlation information. To get the final feature representation $f_\mathbf{x}$ for the input sequence, a mean-pooling operation is leveraged over the output hidden state sequence $h_\mathbf{x}$. Since there exists difference between the textual structures of the source text

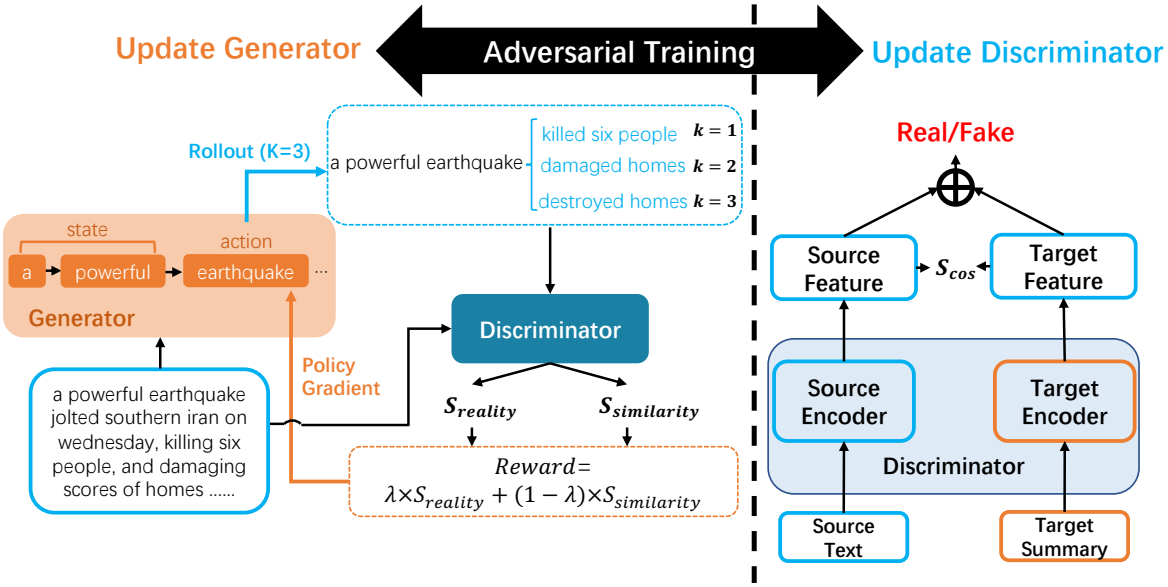

Figure 3: Overview of the model. Left panel: our generator G produces a summary conditioned on the source text. At each time step, the expected reward of a newly generated word ("earthquake" in the presented example) is computed from the siamese semantic-preserving discriminator D using Monte Carlo rollout. We use policy gradient to update the generator G toward generating summaries with higher rewards. Right panel: the discriminator D observes the generated summary and aims at distinguishing it from the real one. Besides, the discriminator D is responsible for capturing the semantic consistency between the source text and the target summary. During adversarial training, both the generator G (left) and the discriminator D (right) are iteratively updated to improve.

and the target summary, for the target summary, we adopt the same encoder framework as the one for the source text, but share no parameters (i.e., pseudo siamese net). And the corresponding final feature representation $f_{\mathbf{y}}$ is also obtained using the mean-pooling operation. Finally, given both the source text and the target summary, the probability that the target summary is classified as real can be calculated as:

$$p = \sigma(V[f_{\mathbf{x}}, f_{\mathbf{y}}]) \tag{3}$$

where V is the weight matrix to transform the concatenation of $f_{\mathbf{x}}$ and $f_{\mathbf{y}}$ into a 2-dimensional embedding and σ is the logistic function. Finally, the training objective for discriminating the real summary from the generated one can be formulated as a supervised classification objective:

$$\mathcal{L}_{real}(\phi) = -\sum_{n=1}^{N} \log p(l^n|\mathbf{x}^n, \mathbf{y}^n; \phi) \tag{4}$$

where N is the number of text-summary pairs, ϕ is the model parameters of the discriminator D, and l^n is the corresponding label (i.e., 0 for the generated summary and 1 for the real summary).

To capture the semantic similarity between the source text and the target summary, we further utilize the final features of the pseudo siamese net. Specifically, we aim to maximize the semantic similarity between the source text and the real summary, while minimizing it for the pair of the source text and the generated summary. To this end, we adopt the cosine function to evaluate the similarity of the sentence pair:

$$\mathcal{S}_{cos} = \frac{\langle f_{\mathbf{x}}, f_{\mathbf{y}} \rangle}{||f_{\mathbf{y}}||\,||f_{\mathbf{y}}||} \tag{5}$$

and the value of $\mathcal{S}_{cos}$ ranges from -1 to 1. Next, we can obtain the contrastive loss $\mathcal{L}_{sim}$ for siamese semantic similarity learning:

$$\begin{aligned}\mathcal{L}_{sim} = \frac{1}{N}\sum_{n=1}^{N} & l^n \mathcal{L}_{+}(f_{\mathbf{x}}^n, f_{\mathbf{y}}^n) \\ & +(1-l^n)\mathcal{L}_{-}(f_{\mathbf{x}}^n, f_{\mathbf{y}}^n)\end{aligned} \tag{6}$$

where N is the number of text-summary pairs, l^n is the corresponding summary label (i.e., 1 for the real sentence pair and 0 for the generated sentence pair), $\mathcal{L}_{+}$ and $\mathcal{L}_{-}$ are the corresponding loss functions for the real and generated sentence pair, respectively. The two sub loss functions are given by:

$$\begin{aligned}\mathcal{L}_{+}(f_{\mathbf{x}}^n, f_{\mathbf{y}}^n) &= (1-\mathcal{S}_{cos})^2 \\ \mathcal{L}_{-}(f_{\mathbf{x}}^n, f_{\mathbf{y}}^n) &= \begin{cases} \mathcal{S}_{cos}^2 & \text{if } \mathcal{S}_{cos} > 0 \\ 0 & \text{otherwise} \end{cases}\end{aligned} \tag{7}$$

Thus, we can obtain the final objective of the siamese semantic-preserving discriminator D:

$$\mathcal{L}_d = \eta \mathcal{L}_{real} + (1-\eta)\mathcal{L}_{sim} \tag{8}$$

where η is a hyper-parameter to balance the two sub training objectives.

3.3 Policy Gradient Training

Following (Yu et al., 2017), during adversarial training, the goal of the generator G is defined as to generate a summary sequence from the start state to maximize its expected overall reward. Formally, the objective function is calculated as:

$$\mathcal{J}_{adv}(\theta) = \sum_{\mathbf{y}_{1:T}} G_\theta(\mathbf{y}_{1:T}|\mathbf{x}) \cdot R_{D_\phi}^{G_\theta}(\mathbf{y}_{1:T-1}, \mathbf{x}, y_T) \tag{9}$$

where θ denotes the parameters of G, $\mathbf{y}_{1:T} = \{y_1, \ldots, y_T\}$ indicates the generated target summary, $\mathbf{x}$ is the source text. Here we denote T_y as T for simplicity. $R_{D_\phi}^{G_\theta}$ is the action-value function of the generated summary given the source text $\mathbf{x}$ (i.e., the expected accumulative reward starting from the state $(\mathbf{y}_{1:T-1}, \mathbf{x})$, taking action y_T, and adopting the policy G_θ). To estimate the action-value function, we combine the probability of being classified as real by the discriminator D with the cosine similarity as the total reward:

$$\begin{aligned} R_{D_\phi}^{G_\theta}(\mathbf{y}_{1:T-1}, \mathbf{x}, y_T) =& \lambda \cdot s_{reality} \\ &+ (1-\lambda) \cdot s_{similarity} \\ s_{reality} =& D_\phi(\mathbf{x}, \mathbf{y}_{1:T}) - b(\mathbf{x}, \mathbf{y}_{1:T}) \\ s_{similarity} =& \mathcal{S}_{cos}(\mathbf{x}, \mathbf{y}_{1:T}) \end{aligned} \tag{10}$$

where $b(\mathbf{x}, \mathbf{y}_{1:T})$ denotes the baseline value to reduce the variance of the reward. In practice, we set it to 0.5 during training. And λ is a hyper-parameter for balance. It is worth noting that, (10) only defines a reward value for a completely generated summary. If $\mathbf{y}_{1:T}$ is partially generated, the values of $D_\phi(\mathbf{x}, \mathbf{y}_{1:T})$ and $\mathcal{S}_{cos}(\mathbf{x}, \mathbf{y}_{1:T})$ are meaningless. To evaluate the action-value for an intermediate state, we apply Monte Carlo (MC) tree search under the policy G_θ to sample the following unknown tokens. Each search lasts until the end of summary token is sampled or the sampled summary reaches the maximum length. For more stable reward and lower variance, we conduct a K-time roll-out as follow:

$$\{\mathbf{y}_{1:T_1}^1, \ldots, \mathbf{y}_{1:T_K}^K\} = MC^{G_\theta}((\mathbf{y}_{1:t}, \mathbf{x}), K) \tag{11}$$

where T_i denotes the length of the summary sampled by the i-th Monte Carlo search. $(\mathbf{y}_{1:t}, \mathbf{x})$ is the current state and $\mathbf{y}_{t+1:T_i}^i$ is sampled based on the policy G_θ. Accordingly, the discriminator provides K rewards for the sampled K summaries respectively. The final reward for the intermediate state is computed as the average of K rewards. Thus, for the generated summary with length T, we compute the final reward for y_t at the sentence level as:

$$R_{D_\phi}^{G_\theta}(\mathbf{y}_{1:t-1}, \mathbf{x}, y_t) = \begin{cases} \frac{1}{K}\sum_{k=1}^{K} \lambda(D_\phi(\mathbf{x}, \mathbf{y}_{1:T}^k) - b(\mathbf{x}, \mathbf{y}_{1:T}^k)) + \\ (1-\lambda)\mathcal{S}_{cos}(\mathbf{x}, \mathbf{y}_{1:T}^k) & t < T \\ \lambda(D_\phi(\mathbf{x}, \mathbf{y}_{1:t}) - b(\mathbf{x}, \mathbf{y}_{1:t})) + \\ (1-\lambda)\mathcal{S}_{cos}(\mathbf{x}, \mathbf{y}_{1:t}) & t = T \end{cases} \tag{12}$$

Using the discriminator D as a reward function can further improve the generator iteratively by dynamically updating D. Once we have a set of more realistic generated summaries, we shall re-train the discriminator model by minimizing (8). Each time when a new discriminator model is obtained, we can re-train the generator. The gradient of the objective $\mathcal{J}_{adv}(\theta)$ w.r.t. the generator's parameters θ can be formulated as:

$$\begin{aligned} \nabla \mathcal{J}_{adv}(\theta) =& \frac{1}{T}\sum_{t=1}^{T}\sum_{y_t} R_{D_\phi}^{G_\theta}(\mathbf{y}_{1:t-1}, \mathbf{x}, y_t) \\ & \cdot \nabla_\theta(G_\theta(y_t|\mathbf{y}_{1:t-1}, \mathbf{x})) \\ =& \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{y_t \in G_\theta}[R_{D_\phi}^{G_\theta}(\mathbf{y}_{1:t-1}, \mathbf{x}, y_t) \\ & \cdot \nabla_\theta \log p(y_t|\mathbf{y}_{1:t-1}, \mathbf{x})] \end{aligned} \tag{13}$$

3.4 Adversarial Training

The overall training flow of SSPGAN is shown in Figure 3. Both the generator G and the siamese semantic-preserving discriminator D learn together by pursuing competing goals. Given $\mathbf{x}$, the generator G generates a summary $\mathbf{y}$. It would prefer summaries with bigger rewards, which implies larger values of $s_{reality}$ and $s_{similarity}$. In contrast, the discriminator D would encourage smaller values of $s_{reality}$ and $s_{similarity}$. Thus, the generator G and the siamese semantic-preserving discriminator D play a min-max game (see Algorithm 1 in the Appendix A.2 for more details).

4 Experiments

4.1 Datasets

We conduct extensive experiments on both Chinese and English text summarization datasets. The Chinese dataset we adopt is a large corpus of Chinese short text summarization (LCSTS) (Hu et al., 2015), which is collected from Sina Weibo, a famous Chinese social media website. Following the data split of previous works, we get around 2.4M text-summary pairs for training, 10K pairs for validation and 725 pairs with annotation score no less than 3 for testing. For English text summarization, we use the Gigaword dataset based on Annotated Gigaword (Napoles et al., 2012), and preprocess it identically to (Rush et al., 2015), which results in 3.8M sentence pairs for training, 190K for validation and around 1.9K for testing.

4.2 Evaluation Metrics

For a fair comparison with previous works, we adopt ROUGE (Lin, 2004) as the automatic evaluation metric. ROUGE measures the degree of overlap between the generated summary and the reference, with respect to the number of n-grams. We report ROUGE-1 (uni-gram), ROUGE-2 (bi-gram), ROUGE-L (longest common subsequence - LCS) on the testing set for our quantitative experiments. Since the official ROUGE evaluation package is only available for English summarization, to evaluate the models on the Chinese summarization task, we follow (Hu et al., 2015) and map all characters including punctuation and numbers to numerical IDs, and then conduct evaluation on them. In experiments, we denote ROUGE as RG for simplicity.

4.3 Compared Models

Baselines for the Chinese text summarization task include the followings. RNN and RNN-context are two RNN-based models adopted in (Hu et al., 2015), without and with the attention mechanism respectively. CopyNet leverages the copy mechanism to alleviate the OOV problem (Gu et al., 2016). RNN-MRT (Shen et al., 2016) and Actor-Critic (Li et al., 2018) are two sentence-level training methods to address the problem of teacher forcing which use the maximum likelihood estimation. DRGD (Li et al., 2017b) uses a recurrent latent random model to strengthen the abstractive text summarization model. GlobalEncoding (Lin et al., 2018) controls the information flow from the encoder to the decoder based on the source-side global information.

System	RG-1	RG-2	RG-L
ABS	29.55	11.32	26.42
ABS+	29.76	11.88	26.96
Concept-pointer+DS	37.01	17.10	34.87
DRGD	36.27	17.57	33.62
Actor-Critic	36.05	17.35	33.49
Transformer	37.57	18.90	34.69
SSPGAN	**38.31**	**19.89**	**35.60**

Table 1: The full-length F-1 based ROUGE scores on the testing set of the English benchmark Gigaword. Here we bold the best results.

As for the English dataset, besides DRGD and Actor-Critic, we choose the following baselines. ABS and ABS+ are two pioneer methods using neural networks for abstractive text summarization (Rush et al., 2015). Concept-pointer+DS engages abstractive summarization models to generate new conceptual words (Wang et al., 2019).

Our model is complemented based on Tensor2Tensor [1]. For all experiments, SSPGAN is run with 5 random seeds on 2 NVIDIA V100 GPUs and the final automatic results are presented with means (see the Appendix A.1 for more details).

4.4 Quantitative Results

4.4.1 English Results

Table 1 shows the results on the English dataset Gigaword. The results of the baselines are reported in the upper rows, while the bottom row summarizes the results of the proposed SSPGAN. When we introduce the SSPGAN framework to Transformer, it significantly improves the performance, proving the effectiveness of our method.

4.4.2 Chinese Results

The experimental results on the Chinese dataset LCSTS are presented in Table 2. As can be observed from the comparison between the baselines in the upper rows and SSPGAN in the bottom row, the proposed method achieves the best performance. In addition, the proposed SSPGAN brings significant improvements to the classical baseline Transformer. Precisely, Transformer is greatly improved in ROUGE-1/2/L with gains of +1.88/+1.09/+1.20.

[1] https://github.com/tensorflow/tensor2tensor

System	RG-1	RG-2	RG-L
RNN	21.50	8.90	18.60
RNN-context	29.90	17.40	27.20
CopyNet	34.40	21.60	31.30
RNN-MRT	37.87	25.43	35.33
Actor-Critic	37.51	24.68	35.02
DRGD	36.99	24.15	34.21
GlobalEncoding	39.40	26.90	36.50
Transformer	42.35	29.38	39.23
SSPGAN	**44.23**	**30.47**	**40.43**

Table 2: The full-length F-1 based ROUGE scores on the testing set of the Chinese benchmark LCSTS. Here we bold the best results.

System	RG-1	RG-2	RG-3
Transformer	37.57	18.90	34.69
+SSPGAN (η, λ=1.0)	38.00	19.41	35.18
+SSPGAN (η, λ=0.7)	**38.31**	**19.89**	**35.60**
+SSPGAN (η, λ=0)	37.78	19.19	34.99

Table 3: Ablation study regarding the sub training objectives proposed in (8) and (10). We bold the best results.

4.5 Analysis

In this section, we analyze the effectiveness of the proposed method from multiple perspectives. All the experiments are conducted on Gigaword.

4.5.1 Ablation Study

As shown in Table 3, we analyze the contributions of different sub training objectives proposed in (8) and (10). On the Transformer model, the basic GAN (i.e., the second row with η=1.0 and λ=1.0) achieves improvement with gains of +0.43/+0.51/+0.49 in ROUGE scores. We also test the results when Transformer is only guided by the semantic similarity objective (i.e., the fourth row with η=0 and λ=0), resulting gains of +0.21/+0.29/+0.30. Armed with the proposed SSPGAN (i.e., the third row with η=0.7 and λ=0.7), the performance can be more significantly improved with gains of +0.74/+0.99/+0.91 in ROUGE scores.

4.5.2 Human Evaluation

To further evaluate the quality of the generated summaries, we randomly select 50 test examples from the Gigaword testing set for human evaluation. For each example, we show the source text, the ground truth summary as well as the summaries generated by different models. The human evaluators do not know which summary comes from which model or which one is the ground truth. Two scores from 1 to 10 are assigned to each summary (1 and 10 indicate the worst and the best respectively), one for readability (how well-written the summary is) and one for consistency (how well the summary conveys the key content of the source text). Each summary is rated by 10 invited human evaluators who are capable of reading English proficiently. And the results are averaged across all selected examples and evaluators. As shown in Table 4, equipped with the basic GAN objective, the readability is improved significantly with comparable results (i.e., the second row with η=1.0 and λ=1.0 and the third row with η=0.7 and λ=0.7). As for the consistency, our proposed model (i.e., the third row with η=0.7 and λ=0.7) achieves the highest score, which justifies that the proposed method can preserve the key content of the source text more accurately. It is worth noting that the improvements of the fourth row are limited, which is only equipped with siamese similarity ob-

System	R	C
Transformer	6.39	6.65
+SSPGAN (η, λ=1.0)	**7.09**	6.82
+SSPGAN (η, λ=0.7)	7.06	**7.33**
+SSPGAN (η, λ=0)	6.64	6.78

Table 4: Comparison of human evaluation on a random subset of the Gigaword testing set. We denote the readability and consistency as R and C, respectively. The best results are bold.

Source: malaysia's national car maker proton **expects to export its cars to russia** by early next year to boost its overseas sales, a company official said tuesday
Reference: malaysian carmaker proton **seeks inroads into russia** by early next year
GAN: malaysia's car maker to boost overseas sales
SSPGAN: malaysia's proton **to export cars to russia**

Source: chinese vice-premier wu yi said tuesday that the country should step up efforts to **develop its service trade** in a bid to alter the growth pattern of foreign trade and increase employment and domestic consumption.
Reference: chinese vice-premier calls for fast **development of service trade**
GAN: chinese vice-premier urges to increase domestic consumption
SSPGAN: chinese vice-premier urges **development of service trade**

Figure 4: Comparison of the summaries generated by the basic GAN and the proposed SSPGAN. Bold text represents that the correct contents are extracted, while the underlined parts correspond to the wrong ones.

jective. Due to the lack of basic GAN objective, the improvement of readability is limited, resulting in incomplete sentence semantic expression and damage to the improvement of consistency.

4.5.3 Case Study

Figure 4 shows some examples of the generated summaries on the English dataset, in which both the basic GAN and the proposed SSPGAN produce readable results. However, as shown in the highlights of the SSPGAN examples, the proposed method is able to convey the key content of the source text more accurately, resulting in more salient summaries as expected. Specifically, in the upper example, the key content "expects to export its car to russia" in the source text is only expressed by SSPGAN, while the basic GAN generates "boost overseas sales", ignoring the most relevant information. Similar behaviors can also be observed in the bottom example.

5 Conclusion

This paper presents a novel siamese generative adversarial net (SSPGAN) which can preserve the semantic consistency between the source text and the target summary for abstractive text summarization. In SSPGAN, a novel semantic similarity based reward is introduced to further augment the GAN-based abstractive text summarization to preserve the semantic consistency and convey the key content in the source text. It is worth noting that SSPGAN addresses the problem of saliency for text summarization from a totally different perspective of semantic consistency, therefore it is orthogonal to some state-of-the-art methods which focus on attention mechanism, and can be applied to them for further improvements.

Acknowledgements

This research was supported by Anhui Provincial Natural Science Foundation (2008085J31) and National Natural Science Foundation of China under No. 62172382.

References

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. 2017. Generalization and equilibrium in generative adversarial nets (gans). In *International Conference on Machine Learning*, pages 224–232. PMLR.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 318–325, Hong Kong. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. 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*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK. Coling 2008 Organizing Committee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. Technical report, MARYLAND UNIV COLLEGE PARK INST FOR ADVANCED COMPUTER STUDIES.

Xiangyu Duan, Hongfei Yu, Mingming Yin, Min Zhang, Weihua Luo, and Yue Zhang. 2019. Contrastive attention mechanism for abstractive sentence summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3044–3053, Hong Kong, China. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Min Gui, Junfeng Tian, Rui Wang, and Zhenglu Yang. 2019. Attention optimization for abstractive document summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1222–1228, Hong Kong, China. Association for Computational Linguistics.

Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. 2017. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *Computing Research Repository*.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale Chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1967–1972, Lisbon, Portugal. Association for Computational Linguistics.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and paste based text summarization. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese CBOW: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 941–951, Berlin, Germany. Association for Computational Linguistics.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization-step one: Sentence compression. *AAAI/IAAI*, 2000:703–710.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169, Copenhagen, Denmark. Association for Computational Linguistics.

Piji Li, Lidong Bing, and Wai Lam. 2018. Actor-critic based training framework for abstractive summarization. *arXiv preprint arXiv:1803.11070*.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017b. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100, Copenhagen, Denmark. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Junyang Lin, Xu Sun, Shuming Ma, and Qi Su. 2018. Global encoding for abstractive summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 163–169, Melbourne, Australia. Association for Computational Linguistics.

Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. 2018. Generative adversarial network for abstractive text summarization. In *Thirty-second AAAI conference on artificial intelligence*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*.

Courtney Napoles, Matthew R Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157.

Joel Larocca Neto, Alex A Freitas, and Celso AA Kaestner. 2002. Automatic text summarization using a machine learning approach. In *Brazilian symposium on artificial intelligence*, pages 205–215. Springer.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *International Conference on Learning Representations*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*

and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shiqi Shen, Yu Zhao, Zhiyuan Liu, Maosong Sun, et al. 2016. Neural headline generation with sentence-wise optimization. *arXiv preprint arXiv:1604.01904*.

Xin Sheng, Linli Xu, Junliang Guo, Jingchang Liu, Ruoyu Zhao, and Yinlong Xu. 2020. Introvnmt: An introspective model for variational neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8830–8837.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wenbo Wang, Yang Gao, Heyan Huang, and Yuxiang Zhou. 2019. Concept pointer network for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3076–3085, Hong Kong, China. Association for Computational Linguistics.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*, pages 112–119.

Junbo Zhao, Michael Mathieu, and Yann LeCun. 2016. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online. Association for Computational Linguistics.

A Appendix

A.1 Experimental Setup

We build both the generator and the siamese semantic-preserving discriminator on the basis of Transformer. For the generator, in both the encoder and the decoder, 6 layers are stacked with dimensions of embedding layers and hidden layers set to 512. The dimension of feed-forward layers is set to 2048. And we set 8 heads for multi-head attention. In the discriminator, both the text encoder and the summary encoder have the same framework as the encoder in the generator, except that the number of layers is set to 2. For the generator, we adopt the joined source-target vocabulary for both English and Chinese experiments. The encoder input embeddings, the decoder input embeddings and the decoder output embeddings are all shared. For the discriminator, the two encoders share the input embeddings.

For the Chinese dataset, we tokenize the sequences into character-level text-summary pairs and evaluate the performance based on the reference tokens. For the English dataset, to improve the computational efficiency and avoid problems with closed vocabularies, we segment the data using byte-pair encoding (BPE) (Sennrich et al., 2016), which results in a vocabulary of 32K tokens.

During pre-training, for the generator, Adam optimizer is used with the learning rate set as 0.0005.

The inverse square root learning rate decay is applied for initial warm up and annealing with 4000 steps. For the discriminator, we adopt RMSProp optimizer with the learning rate of 0.0005 and $\eta = 0.7$. The dropout rate is set to 0.3 for both models. During adversarial training, for both models, the learning rate is set to 0.00001 without changing the optimizer. K in Monte Carlo rollout is set as 20 and λ is 0.7.

In the proposed architecture, there are 2 hyper-parameters η and λ need to be jointly tuned during training. Here we conduct a grid search to find a proper combination of these hyper-parameters. For both η and λ, the value is selected in set $[0.1, 0.3, 0.5, 0.7, 0.9]$ and we experimentally find that the η of 0.7 and the λ of 0.7 give the best results on validation sets.

A.2 Pseudo Code

Algorithm 1 Siamese Semantic-Preserving GAN

Require: generator G_θ, siamese semantic-preserving discriminator D_ϕ, a text summarization dataset $\mathcal{S} = (\mathbf{x}, \hat{\mathbf{y}})$

Initialize G_θ, D_ϕ with random weights θ, ϕ
Pre-train G_θ using (1) on $\mathcal{S}$
Generate negative summaries $\mathbf{y}$ with G_θ for training D
Pre-train D_ϕ using (8) on the combination of $(\mathbf{x}, \mathbf{y})$ and $\mathcal{S}$
while G_θ not converged **do**
  for g-steps **do**
    Generate a sequence $\mathbf{y} = (y_1, \cdots, y_T) \sim G_\theta$
    for t in $1 : T$ **do**
      Calculate $R_{D_\phi}^{G_\theta}(\mathbf{y}_{1:t-1}, \mathbf{x}, y_t)$ using (12)
    end for
    Update generator with policy gradient (13)
  end for
  for d-steps **do**
    Generate negative pairs $(\mathbf{x}, \mathbf{y})$ using latest G_θ and combine them with given positive pairs $\mathcal{S}$
    Train discriminator D_ϕ by (8)
  end for
end while