Paraphrase Generation via Adversarial Penalizations

Anonymous Author(s)
Affiliation
Address
email

Abstract

The paraphrase generation task aims to transform a given sentence in another with the same meaning. It is a very interesting research area because it could be used for almost every Natural Language Processing (NLP) task as a data augmentation technique. Currently, works based on Generative Adversarial Networks (GANs) offer an attractive approach to create synthetic data, especially in images. Although GANs are not originally designed to generate text, some works combine GAN with Reinforcement Learning (RL) obtaining impressive results. In this paper, we propose a novel deep generative model to address the paraphrase generation task. To evaluate the effectiveness of our method, we use the Quora question pairs dataset, which contains duplicated questions. Our proposal outperforms the results of previous baselines.

1 Introduction

In this paper, we propose an adversarial neural network model to address the paraphrase generation task. We observed that the REINFORCE algorithm needs too many update steps to improve. Also, it presents problems of being very fluctuating. Unlike prior approaches [1–4], we use a weighted conditional maximum likelihood to train our generator. So, we take advantage of the maximum likelihood principle, and we can guide the training using a penalization function.

2 Methodology

Our model takes two paired sentences as input. Both sentences express the same idea; however, they use different words and different grammatical structures. The first sentence is the *condition* X, and the second one is the *real* sentence Y. We pre-process both using a tokenizer. We use fastText pre-trained vectors [5] as word representation. We choose fastText over other algorithms due to it is the most recent progress in word embedding algorithms.

2.1 Model Architecture

Our model consists of two networks: generator and discriminator. We define the θ-parameterized generator G_θ to produce a sequence $\hat{Y}_{1:T} = (y_1, y_2, ..., y_T)$ where y_i belongs to a vocabulary. The ϕ-parameterized discriminator D_ϕ is trained to distinguish between the ground-truth and the generator outputs.

Our generator is a Convolutional Sequence to Sequence (ConvS2S) model [6]. We choose this architecture over a Seq2Seq because the ConvS2S needs fewer number of parameters to achieve similar results. That let us to train our framework using large batch sizes to reduce the generator variance [7]. Furthermore, the model performs parallelizable operations to speed up the training time. That feature allowed us to conduct more experiments.

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

The discriminator of our generative model is also based on a ConvS2S architecture. We feed the encoder with the *condition* sentence, and the decoder with either the *generated* or *real* sentence. Different to G_θ, we added two fully connected layers to flatten the output. Finally we pass the result to a sigmoid layer that outputs the probability that the sentence belongs to the *fake* category.

2.2 Training

The *generated* sentence $\hat{Y}$ is the G_θ output. We feed the discriminator with a pair of sentences: *condition-real (real pair)*, or *condition-generated (fake pair)*. Each pair is classified as 0 or 1 in the *real* or *fake* case, respectively. We train D_ϕ using the maximum likelihood principle. On the other hand, we train G_θ using an unified learning objective. We multiply the negative log-likelihood loss of each word by the result of our penalization function. The objective function $J(\theta)$ of G_θ is

$$\nabla J(\theta) = -\sum_{t=1}^{T} \mathbb{E}_{\hat{Y}_{1:T} \sim Y_{1:T}} [\sum_{\hat{y}_t \in \hat{Y}} \log G_\theta(\hat{y}_t | \hat{Y}_{1:T}, X) \cdot P_{D_\phi}^{G_\theta}] \tag{1}$$

We calculate the G_θ log-likelihood loss using the $real$ sentence as decoder input. Nevertheless, we estimate the penalization function $P_{D_\phi}^{G_\theta}$ with the decoding inference result. Thus, we affect the gradients of words that tend to conduct to non-feasible paraphrases.

$P_{D_\phi}^{G_\theta}$ is the discriminator output multiplied by a constant k. The discriminator outputs a score in the interval $[0, 1]$ according to the probability that a sentence is classified as *fake*. That is, sentences classified as fake will receive higher penalizations. However, D_ϕ is trained using complete sequences. To estimate the penalization function in intermediate states, we perform a Monte Carlo search with roll-out G_θ to sample the remaining words. Our penalization function for intermediate timesteps is composed of the average discriminator score of the Monte Carlo samples. We update D_ϕ at each training round to improve the quality of our generated sentences. In addition, it is worth noting that we replace the predicted discriminator penalization by a constant to address the generation of repetitive words. We tuned manually that constant to 1.2.

We present the overall procedure to train our model: as first step, we pre-train G_θ using conditional maximum likelihood with the *condition* and *real* samples. Also we pre-train D_ϕ using supervised learning using pairs composed by *condition-real* or *condition-generated*. Then, we start the adversarial training phase for several rounds: first, we sample and calculate $P_{D_\phi}^{G_\theta}$ to train G_θ using equation 1. After updating the parameters, we output a *generated* sample per *condition* sentence using G_θ. That results in a balanced set of fake and real pairs to feed D_ϕ. Finally, we re-train D_ϕ.

3 Results

We built three sets from the Quora question pairs dataset: Quora I, Quora II, and Quora III. In Quora I, we randomly selected *100K* pairs of sentences for training, *30K* for testing, and *3K* for validation. We did similar in Quora II, but using *50K* sentences for training. In Quora III, the testing set contains *30K* pairs which questions do not appear in the training (*50K*) and validation (*3K*) sets. That makes Quora III the most challenging set. Table 1 shows results for Quora I, II, and III corpora. The results presented refer to the BLEU scores of testing sets. The state-of-the-art approach is marked with (*), and our results are in **bold**.

Table 1: Comparative results on datasets Quora I, II, and III.

Proposed	BLEU (up 2-grams)		BLEU (up 4-grams)	
Model	Quora I	Quora III	Quora I	Quora II
VAE-SVG[8]	-	-	22.50	17.10
VAE-SVG-eq[8]	-	-	22.90	17.40
RbM-SL*[9]	43.54	35.81	-	-
RbM-IRL*[9]	43.09	34.79	-	-
ConvS2S[6]	43.07	37.74	46.76	45.37
Para-GAN	**43.66**	**39.62**	**47.03**	**40.11**

References

[1] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

[2] William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: Better text generation via filling in the _. *arXiv preprint arXiv:1801.07736*, 2018.

[3] Yang Li, Quan Pan, Suhang Wang, Tao Yang, and Erik Cambria. A generative model for category text generation. *Information Sciences*, 450:301–315, 2018.

[4] Xinyue Liu, Xiangnan Kong, Lei Liu, and Kuorong Chiang. Treegan: Syntax-aware sequence generation with generative adversarial networks. *arXiv preprint arXiv:1808.07582*, 2018.

[5] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[6] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

[7] Cyprien de Masson d'Autume, Mihaela Rosca, Jack Rae, and Shakir Mohamed. Training language gans from scratch. *arXiv preprint arXiv:1905.09922*, 2019.

[8] Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. A deep generative framework for paraphrase generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[9] Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, 2018.