NEURAL MACHINE TRANSLATION USING RECURRENT GENERATIVE ADVERARIAL NETWORKS

A PREPRINT

Nazim Shaikh
Department of Electrical Engineering
Viterbi School of Engineering
University of Southern California
nshaikh@usc.edu

May 7, 2019

ABSTRACT

Adversarial Training or Generative Adversarial Networks have shown great promise in Image Generation and are able to achieve state-of-the-art results. However, For sequential data such as text, training GANs has proven to be difficult. One reason is because of non-differentiable nature of generating text with R NNs. Consequently, there have been past work where GANs have been employed for NLP tasks such as text generation, sequence labelling, etc. As a part of directed research, I examine one such application of GAN with RNN called Adversarial Neural Machine Translation.

1 Introduction

Neural Machine Translation (NMT) is a process of translating a sentence from one language into a language of your choice. NMT is widely used in both academic and industry and has become a part of our day to day life. Despite its success, the translation quality of NMT system is still unsatisfied and there remains a large room for improvement. The current NMT model aims to maximize the probability of the ground-truth sentence given source sentence. Such an objective does not guarantee the translation results to be natural and sufficient like human-translations. Some previous works tried to alleviate this limitation by reducing the objective inconsistency between NMT training and inference. Some improvement is observed, but the objective still cannot bridge the gap between NMT translations and human-generated translations. Thanks to the success of Generative Adversarial Networks (GANs)[1], several works have been done to adopt GANs for NMT. The latest two works are [2] and [3]. Here, I manage to investigate how NMT system works with Adversarial Training and implement an Adversarial NMT model according to the work of [2].

In Adversarial-NMT, besides the typical NMT model, an adversary is introduced to minimize distinction between the translation generated by NMT from that by human (i.e., ground truth). The NMT model tries to improve its translation results such that it can successfully fool the adversary. These two modules in Adversarial-NMT are jointly trained, and their performances get mutually improved.

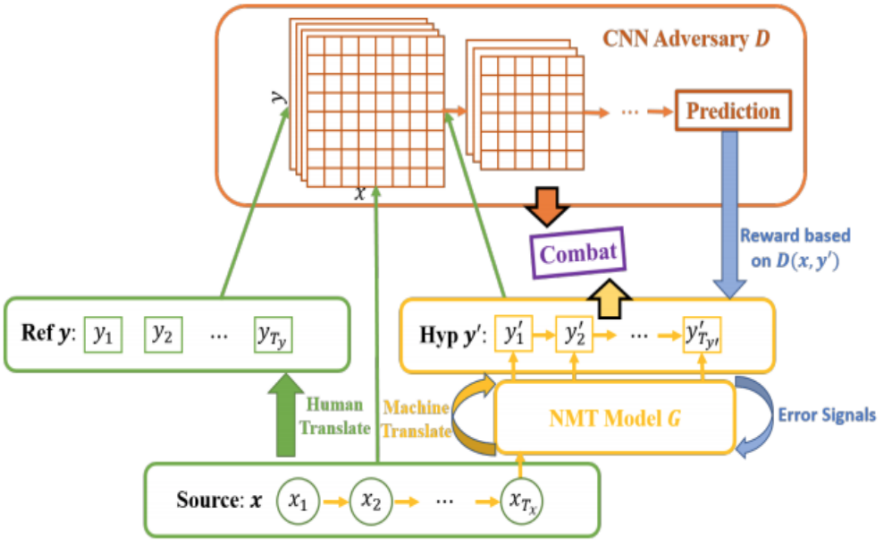
2 Architecture

2.1 Generative Adversarial Networks

Generative adversarial networks (GANs) are a class of neural network architectures designed with the aim of generating realistic data [1]. The approach involves training two neural models with conflicting objectives, one generator (G), and one discriminator (D), forcing each other to improve. The generator tries to produce samples that looks real, and the discriminator tries to discriminate between generated samples and real data. Using this framework, makes it possible to train deep generative models without expensive normalizing constants.

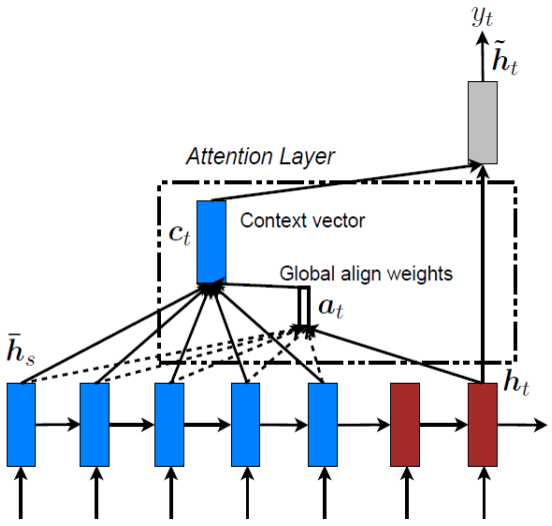
2.2 Adversarial NMT

The overall framework of Adversarial-NMT is shown in Figure 1. Let x and y be a bilingual aligned sentence pair for training, where x_i is the i-th word in the source sentence and y_j is the j-th word of the target sentence. Let y' denote the translation sentence out from an NMT system for the source sentence x. The goal of Adversarial-NMT is to force y' to be as similar as the human translation y. While the goal of the adversary is to differentiate human translation from machine translation, and the generator tries to produce a target sentence as similar as human translation so as to fool the adversary.



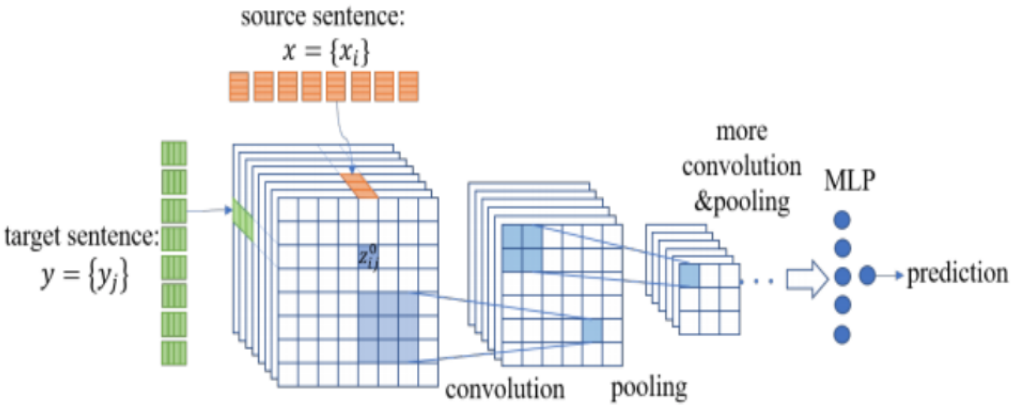
2.2.1 Generator

The task of generator would be generate a translation which is as close to human-level translation. For this purpose, a Recurrent Neural Network (RNN) based encoder-decoder is used as generator model. Additionally, An attention mechanism has been introduced to improve neural machine translation by selectively focusing on parts of the source sentence during translation as specified in paper [4].



2.2.2 Discriminator

The task of discriminator is to measure translative matching degree of the source-target sentence pair. Here, discriminator is a convolutional neural networks, since with its layer-by-layer convolution and pooling strategies, CNN is able to accurately capture the hierarchical correspondence of source-target sentence pairs at different abstraction levels. The



general structure is as shown in the figure. We first concatenate the embedding vectors in the source sentence x and the target sentence y, and each word y_j in the target sentence y will be concatenated to every word in x.This constructs a 2D image-like representation with the image size of the square of the sequence length, and there are 2x the embedding size channels in total for one sample input to the CNN. We then perform convolution on a 3x3 window, with the purpose of capturing the correspondence between segments in source and target sentences pushing the result to the sigmoid activation function. After that we perform a max-pooling in non-overlapping 2x2 windows. We go on for two layers of convolutions and max-pooling, aiming at capturing the correspondence at different levels of abstraction, but also avoiding to make the network too complex. The extracted features are then fed into a multi-layer perceptron (MLP) and another fully connected layer to output a vector with only two elements corresponding to the two classes (human-translated sentence and generated sentence) respectively, with sigmoid activation at the last layer to give the probability indicating how much similar or different the sentence pair is from ground-truth data. The optimization target of such a CNN adversary is to minimize the cross entropy loss for binary classification, with ground data as positive instances while sampled data as negative instances.

2.3 Training

We use policy gradient algorithm to train Adversarial-NMT, to tackle the problem that the discretely sampled y' from the NMT model G makes it difficult to directly back-propagate the error signals from the adversary model D to G. For the convenience of explanation, let $p_{data}(x)$ represent the distribution of true translation data, let p_g represent the generator's distribution over data, while $p_x(z)$ is the distribution of noise z, which is the randomly initialized word embedding. Let $G(z)$ be the output of the generator, and $D(z)$ is the probability that x is the ground-truth translation rather than data produced by the generator.

For the discriminator D, it is trained to better classify real translation from the fake translation from the generator G. So it should tend to output 1 for true translation and 0 for fake translation, as its output is the probability that the sentence is human-translated. That is,

$$\text{if } x\ p_{data}(x), D(x) = 1, E_{x\ P_{data}(x)} log(D(x)) \text{ is the maximum}$$
$$\text{if } x\ p_z(z), D(G(z)) = 0, E_{x\ P_z(z)} log(1 - D(G(z)) \text{ is the maximum}$$

As per [1] the value function is defined as

$$V(G, D) = log(D(x)) + log(1 - D(G(z))$$

Here, the term $log(1 - D(G(z))$ provided by the adversary D, acts as a Monte-Carlo estimation of the reward.

The best discriminator is given by

$$D_G^* = argmaxV(G, D)$$

While the goal of the generator G is to fool the discriminator D that the sentence it generated is true translation. So the best generator should be

$$G_D^* = argmin(argmaxV(G, D))$$

It can be proven the best generator exists. Based on [2], following loss is used for the training of the two models. We use cross entropy as the criterion. G aims to let D believe that the generated translation is true data, thus the loss is

$$L_G = H(1, D(G(z)))$$

And the loss for training the discriminator is

$$L_G = H(1, D(x)) + H(0, D(G(z)))$$

The above equations imply that the more likely y' to successfully fool D, i.e. larger L_D, the larger reward the NMT model will get, and the fake training data (x, y') will correspondingly be more favored to improve the G.

3 Experiments

3.1 Data

The model was trained and tested on 2 different translation datasets. The dataset used is different than one used in [2]. First, For German->English translation, the dataset is from IWSLT 2014 evaluation consisting of training/dev/test corpus with approximately 160k, 7.2k and 6.7k bilingual sentence pairs respectively. Second, For Czech->English translation, the dataset is from IWSLT 2015 evaluation consisting of training/dev/test corpus with approximately 93k, 4.2k and 5.7k bilingual sentence pairs respectively. Both the datasets were preprocessed using [5] - fairseq, a sequence modelling toolkit build by facebook AI research team.

3.2 Implementation Details

The implementation is little different compared to what mentioned in the paper [2]. The generator model is a double layer LSTMs acting as encoder and decoder. The dimensions of word embedding and LSTM hidden state are respectively set to 1024. For the adversary D, the CNN consists of two convolution+pooling layers, 2 MLP layer and one softmax layer, with 3 x 3 convolution window size, 2 x 2 pooling window size, 20 feature map size and 20 MLP hidden layer size.

3.3 Training

The generator and discriminator are trained together. The source sequence batch is passed through generator to generate fake translations. The generated fake translations and the ground-truth translations are then sent to the discriminator. Different loss is calculated for G and D. First, the parameters of discriminator are updated by backpropagation and then of the generator. 50% of randomly chosen mini-batch data is trained with policy gradient based training, while MLE principle(negative log-likelihood loss) is applied to the other mini-batch data. The MLE acts as a regularizer to guarantee smooth model update, alleviating the negative effects brought by high gradient estimation variance introduced by the policy gradient algorithm. Acting this way, significantly improves model stability

I used batch size of 64, learning rate of 0.001 and Adam optimizer to train our model. Training was done for 10 epochs and took 4hrs to complete.Additionally, for comparison, same setting was applied to train the NMT model alone. The final translation quality was measured by tokenized case sensitive BLEU score [6]

3.4 Results

In order to show that Adversarial-NMT has better performance than original NMT models, we compare its performance with the trained-alone generator with same experiment settings as mentioned above. Table 1 shows BLEU score for different translation tasks evaluated on 2 systems. From the table, we can clearly observe that Adversarial-NMT obtains satisfactory translation quality against baseline system for both De -> en and Cs -> En translation task.

Additionally, to better visualize and understand the advantages of adversarial training brought by Adversarial-NMT, Table 2 show sample illustration of translation quality using baseline and adversarial-NMT emphasized on their different parts by bold fonts leading to different translation quality. It can be seen that the translation quality of adversarial-NMT is more both in terms of subjective feelings and BLEU scores as a quantitative measure.

System	BLEU (De -> En)	BLEU (Cs -> En)
Attention based NMT	23.87	22.7
Adversarial NMT	27.5	25.8

Table 1: NMT systems performance

Source Sentence	Es fühlt, was ich tun möchte, wohin ich gehen möchte, um meine Kräfte zu sammeln und sie zu erhalten.	BLEU
Ground Truth Translation	It feels what i want to do , where i want to go and pick up my strength and sustain it .	
Translation by Attention based NMT	It **feels** what i want do , where i want to go and **collect strength and endure** .	48.51
Translation by Adversarial NMT	It **senses** what i want to do , where i want to go and **then augments my strength and endurance** .	57.18

Table 2: sample example demonstrating the translation quality improvement brought by Adversarial-NMT

4 Conclusion

Although this approach does not generate state of the art results, addition of adversary to the original NMT model helps generate more natural, sufficient and accurate translation, and directly minimize the difference between machine and human translation. I investigated the idea and the implementation of GANs in detail, and build an architecture similar to that of Adversarial-NMT model using the [7] github repo as support. I then conducted some experiments on this model using the German->English and Czech->English translation dataset, and the results on test set imply that Adversarial-NMT model can produce a better translation than the original NMT model. Few other observations made during are :- the model seems to perform fairly well for sentences less than 35 words, also the discriminator seems to overfit pretty early. This may be due to data to parameter ratio. It may have too many parameters.

5 Future Work

There are many improvements that can be done. since the discriminator overfits easily, it can be reconstructed to a simpler network, with fewer layers or even without the convolution and pooling layers. Next, discriminator can be initially pre-trained using the sampled data sampled from the NMT model and the ground-truth translation. We can also pre-train the generator independently. Parameters can be further tuned in both generator and discriminator model. The same model can be trained for sentences longer than 50 words in length in order to check the feasibility with long sentence translations

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv*, 2014.

[2] Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yun Liu. Adversarial neural machine translation. *arXiv*, 2017.

[3] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Improving neural machine translation with conditional sequence generative adversarial nets. *arXiv*, 2017.

[4] Minh-Thang Luong, Hieu Pham, and Christopher Manning. Effective approaches to attention-based neural machine translation. *arXiv*, 2015.

[5] fairseq. `https://github.com/pytorch/fairseq`.

[6] mosesdecoder. `https://github.com/moses-smt/mosesdecoder`.

[7] Adversarial-nmt. `https://github.com/wangyirui/Adversarial-NMT`.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv*, 2014.

[9] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: sequence generative adversarial nets with policy gradient. *arXiv*, 2016.

[10] Mauro Cettolo, Christian Girardi, and Marcello Federico. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, May 2012.