Generating Text through Adversarial Training using Skip-Thought Vectors

Afroz Ahamad
Department of Computer Science and Information Systems,
Birla Institute of Technology and Science, Pilani
afrozsahamad@gmail.com

Abstract

In the past few years, various advancements have been made in generative models owing to the formulation of Generative Adversarial Networks (GANs). GANs have been shown to perform exceedingly well on a wide variety of tasks pertaining to image generation and style transfer. In the field of Natural Language Processing, word embeddings such as word2vec and GLoVe are state-of-the-art methods for applying neural network models on textual data. Attempts have been made for utilizing GANs with word embeddings for text generation. This work presents an approach to text generation using Skip-Thought sentence embeddings in conjunction with GANs based on gradient penalty functions and f-measures. The results of using sentence embeddings with GANs for generating text conditioned on input information are comparable to the approaches where word embeddings are used.

1. Introduction

Numerous efforts have been made in the field of natural language text generation for tasks such as sentiment analysis (Zhang et al., 2018) and machine translation (Gangi & Federico; Qian et al., 2018). Early techniques for generating text conditioned on some input information were template or rule-based engines, or probabilistic models such as n-gram. In recent times, state-of-the-art results on these tasks have been achieved by recurrent (Press et al.; Mikolov et al., 2010) and convolutional neural network models trained for likelihood maximization. This work proposes an approach for text generation using Generative Adversarial Networks with Skip-Thought vectors.

GANs (Goodfellow et al., 2014) are a class of neural networks that explicitly train a generator to produce high-quality samples by pitting against an adversarial discriminative model. GANs output differentiable values and hence the task of discrete text generation has to use vectors as differentiable inputs. This is achieved by training the GAN with sentence embedding vectors produced by Skip-Thought (Kiros et al., 2015), a neural network model for learning fixed length representations of sentences.

2. Related Works

Deep neural network architectures have demonstrated strong results on natural language generation tasks (Xie, 2017). Recurrent neural networks using combinations of shared parameter matrices across time-steps (Sutskever et al., 2014; Mikolov et al., 2010; Cho et al., 2014) with different gating mechanisms for easing optimization (Hochreiter & Schmidhuber, 1997; Cho et al., 2014) have found some success in modeling natural language. Another approach is to use convolutional neural networks that reuse kernels across time-steps with attention mechanism to perform language generation tasks (Kalchbrenner et al., 2016; 2014).

Supervised learning with deep neural networks in the framework of encoder-decoder models has become the state-of-the-art methods for approaching NLP problems (Young et al.). Stacked denoising autoencoder have been used for domain adaptation in classifying sentiments (Glorot et al., 2011) and combinatory categorical autoencoders demonstrate learning the compositionality of sentences (Hermann & Blunsom, 2013). Recent text generation models use a wide variety of GANs such as gradient policy based sequence generation framework (Yu et al., 2016) and an actor-critic conditional GAN to fill missing text conditioned on

Code available at:
https://github.com/enigmaeth/skip-thought-gan

surrounding text (Fedus et al., 2018) for performing natural language generation tasks. Other architectures such as those proposed in (Wang et al., 2017) with RNN and variational auto-encoder generator with CNN discriminator and in (Guo et al., 2017) with leaky discriminator to guide generator through high-level extracted features have also shown great results.

3. Skip-Thought Generative Adversarial Network (STGAN)

This section introduces Skip-Thought Generative Adversarial Network with a background on models that it is based on. The Skip-Thought model (Kiros et al., 2015) induces embedding vectors for sentences present in training corpus. These vectors constitute the real distribution for the discriminator network. The generator network produces sentence vectors similar to those from the encoded real distribution. The generated vectors are sampled over training and decoded to produce sentences using a Skip-Thought decoder conditioned on the same text corpus.

3.1. Skip-Thought Vectors

Skip-Thought is an encoder-decoder framework with an unsupervised approach to train a generic, distributed sentence encoder. The encoder maps sentences sharing semantic and syntactic properties to similar vector representations and the decoder reconstructs the surrounding sentences of an encoded passage. The sentence encoding approach draws inspiration from the skip-gram model in producing vector representations using previous and next sentences.

The Skip-Thought model uses an RNN encoder with GRU activations (Chung et al., 2014) and an RNN decoder with conditional GRU, the combination being identical to the RNN encoder-decoder of (Cho et al., 2014) used in neural machine translation.

3.1.1. Skip-Thought Architecture

For a given sentence tuple (s_{i-1}, s_i, s_{i+1}), let $\mathbf{w}_i^t$ denote the t-th word for sentence $\mathbf{s}_i$, and let $\mathbf{x}_i^t$ denote its word embedding. The model has three components:
Encoder. Encoded vectors for a sentence s_i with N words w^i, w^{i+1},...,w^n are computed by iterating over the following sequence of equations:

$$\begin{aligned}\mathbf{r^t} &= \sigma(\mathbf{W_r x^t} + \mathbf{U_r h^{t-1}})\\ \mathbf{z^t} &= \sigma(\mathbf{W_z x^t} + \mathbf{U_z h^{t-1}})\\ \hbar^{\mathbf{t}} &= \tanh(\mathbf{W x^t} + \mathbf{U}(\mathbf{r^t} \odot \mathbf{h^{t-1}}))\\ \mathbf{h^t} &= (\mathbf{1} - \mathbf{z^t}) \odot \mathbf{h^{t-1}} + \mathbf{z^t} \odot \hbar^{\mathbf{t}}\end{aligned}$$

where h_i^t is a hidden state at each time step and interpreted as a sequence of words w_i^1,...,w_i^n,$\hbar^t$ is the proposed state update at time t, $\mathbf{z}^t$ is the update gate and $\mathbf{r}^t$ is the reset gate. Both update gates take values between zero and one.

Decoder. A neural language model conditioned on the encoder output h_i serves as the decoder. Bias matrices C_z, C_r, C are introduced for the update gate, reset gate and hidden state computation by the encoder. Two decoders are used in parallel, one each for sentences $\mathbf{s}_i + 1$ and $\mathbf{s}_i - 1$. The following equations are iterated over for decoding:

$$\begin{aligned}\mathbf{r^t} &= \sigma(\mathbf{W_r^d x^{t-1}} + \mathbf{U_r^d h^{t-1}} + \mathbf{C_r h_i})\\ \mathbf{z^t} &= \sigma(\mathbf{W_z^d x^{t-1}} + \mathbf{U_z^d h^{t-1}} + \mathbf{C_z h_i})\\ \hbar^{\mathbf{t}} &= \tanh(\mathbf{W^d x^{t-1}} + \mathbf{U^d}(\mathbf{r^t} \odot \mathbf{h^{t-1}}) + \mathbf{C h_i})\\ \mathbf{h_{i+1}^t} &= (\mathbf{1} - \mathbf{z^t}) \odot \mathbf{h^{t-1}} + \mathbf{z^t} \odot \hbar^{\mathbf{t}}\end{aligned}$$

Objective. For the same tuple of sentences, objective function is the sum of log-probabilities for the forward and backward sentences conditioned on the encoder representation:

$$\sum_t \mathbf{log} P(w_{i+1}^t | w_{i+1}^{<t}, h_i) + \sum_t \mathbf{log} P(w_{i-1}^t | w_{i-1}^{<t}, h_i)$$

3.2. Generative Adversarial Networks

Generative Adversarial Networks (Goodfellow et al., 2014) are deep neural net architectures comprised of two networks, contesting with each other in a zero-sum game framework. For a given data, GANs can mimic learning the underlying distribution and generate artificial data samples similar to those from the real distribution. Generative Adversarial Networks consists of two players - a Generator and a Discriminator. The generator G tries to produce data close to the real distribution $P(x)$ from some stochastic distribution $P(z)$ termed as noise. The discriminator D's objective is to differentiate between real and generated data $G(z)$.

The two networks - generator and discriminator compete against each other in a zero-sum game. The minimax strategy dictates that each network plays optimally with the assumption that the other network is optimal. This leads to Nash equilibrium which is the point of convergence for GAN model.

Objective. (Goodfellow et al., 2014) have formulated the minimax game for a generator G, discriminator D adversarial network with value function $V(G, D)$ as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\, log D(x)\,] + \mathbb{E}_{z \sim p_z(z)}[\, log\,(1 - D(G(z)))\,]$$

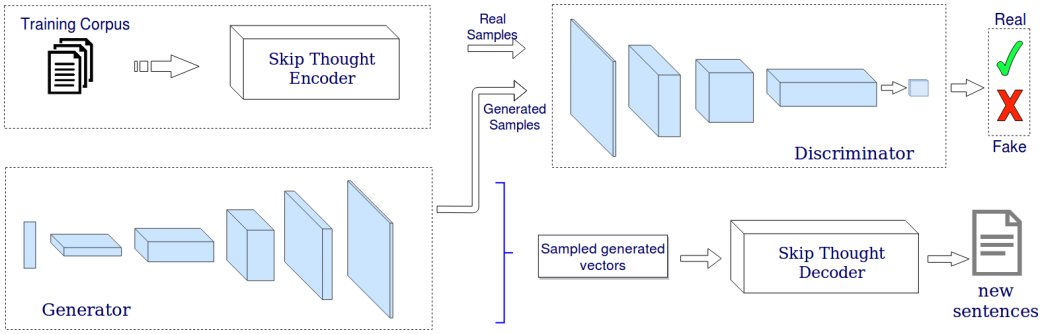

Figure 1. Skip-Thought Generative Adversarial Network model architecture

3.3. Model Architecture

The STGAN model uses a deep convolutional generative adversarial network, similar to the one used in (Radford et al.). The generator network is updated twice for each discriminator network update to prevent fast convergence of the discriminator network.

The Skip-Thought encoder for the model encodes sentences with length less than 30 words using 2400 GRU units (Chung et al., 2014) with word vector dimensionality of 620 to produce 4800-dimensional combine-skip vectors. (Kiros et al., 2015). The combine-skip vectors, with the first 2400 dimensions being uni-skip model and the last 2400 bi-skip model, are used as they have been found to be the best performing in the experiments[1]. The decoder uses greedy decoding taking argmax over softmax output distribution for given time-step which acts as input for next time-step. It reconstructs sentences conditioned on a sentence vector by randomly sampling from the predicted distributions with or without a preset beam width. Unknown tokens are not included in the vocabulary. A 620 dimensional RNN word embeddings is used with 1600 hidden GRU decoding units. Gradient clipping with Adam optimizer (Kingma & Ba, 2014) is used, with a batch size of 16 and maximum sentence length of 100 words for decoder.

3.4. Improving Training and Loss

The training process of a GAN is notably difficult (Salimans et al., 2016) and several improvement techniques such as batch normalization, feature matching, historical averaging (Salimans et al., 2016) and unrolling GAN (Metz et al.) have been suggested for making the training more stable. Training the Skip-Thought GAN often results in mode dropping (Arjovsky & Bottou; Srivastava et al.) with a parameter setting where it outputs a very narrow distribution of points. To overcome this, it uses minibatch discrimination by looking at an entire batch of samples and modeling the distance between a given sample and all the other samples present in that batch.

The minimax formulation for an optimal discriminator in a vanilla GAN is Jensen-Shannon Distance between the generated distribution and the real distribution. (Arjovsky et al., 2017) used Wasserstein distance or earth mover's distance to demonstrate how replacing distance measures can improve training loss for GAN. (Gulrajani et al., 2017) have incorporated a gradient penalty regularizer in WGAN objective for discriminator's loss function. The experiments in this work use the above f-measures to improve performance of Skip-Thought GAN on text generation.

4. Results and Discussion

4.1. Conditional Generation of Sentences.

GANs can be conditioned on data attributes to generate samples (Mirza & Osindero, 2014; Radford et al.). In this experiment, both the generator and discriminator are conditioned on Skip-Thought encoded vectors (Kiros et al., 2015). The encoder converts 70000 sentences from the BookCorpus dataset (Zhu et al., 2015) with a training/test/validation split of 5/1/1 into vectors used as real samples for discriminator. The decoded sentences are used to evaluate model performance under corpus level BLEU-2, BLEU-3 and BLEU-4 metrics (Papineni et al.), once using only test set as reference and then entire corpus as reference. Table 1 compares these results for different architectures that have been experimented with in this paper.

[1] https://github.com/ryankiros/skip-thoughts/

Model	Test set	Complete corpus reference		
	BLEU-3	BLEU-2	BLEU-3	BLEU-4
STGAN	0.521	**0.709**	0.564	0.525
STGAN*(minibatch)*	0.526	**0.745**	0.607	0.531
STGAN-GP	0.558	**0.791**	0.621	0.547
STWGAN	0.582	**0.833**	0.669	0.580
STWGAN-GP	**0.617**	**0.836**	**0.682**	**0.594**

Table 1. BLEU-2, BLEU-3 and BLEU-4 metric scores for different models with **(a)** test set as reference, and **(b)** entire corpus as reference. **ST**: Skip-Thought, **GAN**: Generative Adversarial Network, **W**: Wasserstein

Mode collapse	With minibatch discrimination	With gradient penalty
it ?	it a bottle ?	battery is eighteen percent um ?
it ?	a glass bottle ?	what fine are cash please ?
it ?	a glass bottle it ?	you're gonna go over the t- house .
it ? how would it ?	it my hand a bottle ?	do you have a nice store around here?
it ? how would it ?	the phone my hand it	open this flight number six zero one.

Wasserstein STGAN	Wasserstein STGAN with gradient penalty
we have new year 's holidays, always.	my passport and a letter card with my card , please
here you can n't see your suitcase ,	here on my telephone, mr. kimura's registration card's address.
please show me how much is a transfer?	i can n't see some shopping happened .
i had a police take watch out of my wallet .	get him my camera found a person 's my watch .
here i collect my telephone card and telephone number	delta airlines flight six zero two from six p.m. to miami, please?

Table 2. Sample sentences generated from training on CMU-SE Dataset; mode collapse is overcome by using minibatch discrimination. Formation of sentences further improved by changing f-measure to Wasserstein distance along with gradient penalty regularizer.

4.2. Language Generation.

Language generation is done on a dataset comprising simple English sentences referred to as CMU-SE[2] in (Rajeswar et al., 2017). The CMU-SE dataset consists of 44,016 sentences with a vocabulary of 3,122 words. For encoding, the vectors are extracted in batches of sentences having the same length. The samples represent how mode collapse is manifested when using least-squares distance (Mao et al., 2016) f-measure without minibatch discrimination. Table 2(a) contains sentences generated from STGAN using least-squares distance (Mao et al., 2016) in which there was no mode collapse observed, while 2(b) contains examples wherein it is observed. Table 2(c) shows generated sentences using gradient penalty regularizer(GAN-GP). Table 2(d) has samples generated from STGAN when using Wasserstein distance f-measure as WGAN (Arjovsky et al., 2017)) and 2(e) contains samples when using a gradient penalty regularizer term as WGAN-GP (Gulrajani et al., 2017).

[2]https://github.com/clab/sp2016.11-731/tree/master/hw4/data

4.3. Further Work

Another performance metric that can be computed for this setup has been described in (Rajeswar et al., 2017) which is a parallel work to this. Simple CFG[3] and more complex ones like Penn Treebank CFG generate samples (Eisner & Smith, 2008) which are used as input to GAN and the model is evaluated by computing the diversity and accuracy of generated samples conforming to the given CFG.

Skip-Thought sentence embeddings can be used to generate images with GANs conditioned on text vectors for text-to-image conversion tasks like those achieved in (Reed et al.; Bodnar, 2018). These embeddings have also been used to Models like neural-storyteller[4] which use these sentence embeddings can be experimented with generative adversarial networks to generate unique samples.

[3]http://www.cs.jhu.edu/~jason/465/hw-grammar/extra-grammars/holygrail

[4]https://github.com/ryankiros/neural-storyteller

References

Arjovsky, M. and Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. *ArXiv e-prints*.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *ArXiv e-prints*, January 2017.

Bodnar, Cristian. Text to image synthesis using generative adversarial networks. *CoRR*, abs/1805.00676, 2018.

Cho, Kyunghyun, van Merrienboer, Bart, Bahdanau, Dzmitry, and Bengio, Yoshua. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, 2014.

Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Yoshua. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Eisner, Jason and Smith, Noah A. Competitive grammar writing. In *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics*, pp. 97–105. Association for Computational Linguistics, 2008.

Fedus, W., Goodfellow, I., and Dai, A. M. MaskGAN: Better Text Generation via Filling in the_____. *ArXiv e-prints*, January 2018.

Gangi, Mattia Antonino Di and Federico, Marcello. Deep neural machine translation with weakly-recurrent units. *CoRR*, abs/1805.04185.

Glorot, Xavier, Bordes, Antoine, and Bengio, Yoshua. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 513–520, 2011.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron C. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. 2017.

Guo, Jiaxian, Lu, Sidi, Cai, Han, Zhang, Weinan, Yu, Yong, and Wang, Jun. Long text generation via adversarial training with leaked information. *CoRR*, abs/1709.08624, 2017.

Hermann, Karl Moritz and Blunsom, Phil. The role of syntax in vector space models of compositional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 894–904, 2013.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural Comput.*, 9(8), November 1997.

Kalchbrenner, Nal, Grefenstette, Edward, and Blunsom, Phil. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014.

Kalchbrenner, Nal, Espeholt, Lasse, Simonyan, Karen, van den Oord, Aäron, Graves, Alex, and Kavukcuoglu, Koray. Neural machine translation in linear time. *CoRR*, abs/1610.10099, 2016.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Kiros, Ryan, Zhu, Yukun, Salakhutdinov, Ruslan, Zemel, Richard S, Torralba, Antonio, Urtasun, Raquel, and Fidler, Sanja. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*, 2015.

Mao, Xudong, Li, Qing, Xie, Haoran, Lau, Raymond Y. K., and Wang, Zhen. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, 2016.

Metz, Luke, Poole, Ben, Pfau, David, and Sohl-Dickstein, Jascha. Unrolled generative adversarial networks. *CoRR*, abs/1611.02163.

Mikolov, Tomas, Karafiát, Martin, Burget, Lukás, Cernocký, Jan, and Khudanpur, Sanjeev. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

Mirza, Mehdi and Osindero, Simon. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02.

Press, Ofir, Bar, Amir, Bogin, Ben, Berant, Jonathan, and Wolf, Lior. Language generation with recurrent generative adversarial networks without pre-training. *CoRR*, abs/1706.01399.

Qian, Xin, Zhong, Ziyi, and Zhou, Jieli. Multimodal machine translation with reinforcement learning. *CoRR*, abs/1805.02356, 2018.

Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434.

Rajeswar, Sai, Subramanian, Sandeep, Dutil, Francis, Pal, Christopher Joseph, and Courville, Aaron C. Adversarial generation of natural language. *CoRR*, abs/1705.10929, 2017.

Reed, Scott E., Akata, Zeynep, Yan, Xinchen, Logeswaran, Lajanugen, Schiele, Bernt, and Lee, Honglak. Generative adversarial text to image synthesis. *CoRR*, abs/1605.05396.

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 2016.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C. VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning. *ArXiv e-prints*.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

Wang, Heng, Qin, Zengchang, and Wan, Tao. Text generation based on generative adversarial nets with latent variable. *CoRR*, abs/1712.00170, 2017.

Xie, Ziang. Neural text generation: A practical guide. *arXiv preprint arXiv:1711.09534*, 2017.

Young, Tom, Hazarika, Devamanyu, Poria, Soujanya, and Cambria, Erik. Recent trends in deep learning based natural language processing. *CoRR*, abs/1708.02709.

Yu, Lantao, Zhang, Weinan, Wang, Jun, and Yu, Yong. Seqgan: Sequence generative adversarial nets with policy gradient. *CoRR*, abs/1609.05473, 2016.

Zhang, Lei, Wang, Shuai, and Liu, Bing. Deep learning for sentiment analysis : A survey. *CoRR*, abs/1801.07883, 2018.

Zhu, Yukun, Kiros, Ryan, Zemel, Richard S., Salakhutdinov, Ruslan, Urtasun, Raquel, Torralba, Antonio, and Fidler, Sanja. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. 2015.