

Generating Text through Adversarial Training using Skip-Thought Vectors

Afroz Ahamad

BITS Pilani Hyderabad Campus
afroz.sahamad@gmail.com

Abstract

GANs have been shown to perform exceedingly well on tasks pertaining to image generation and style transfer. In the field of language modelling, word embeddings such as GloVe and word2vec are state-of-the-art methods for applying neural network models on textual data. Attempts have been made to utilize GANs with word embeddings for text generation. This study presents an approach to text generation using Skip-Thought sentence embeddings with GANs based on gradient penalty functions and f-measures. The proposed architecture aims to reproduce writing style in the generated text by modelling the way of expression at a sentence level across all the works of an author. Extensive experiments were run in different embedding settings on a variety of tasks including conditional text generation and language generation. The model outperforms baseline text generation networks across several automated evaluation metrics like BLEU-n, METEOR and ROUGE. Further, wide applicability and effectiveness in real life tasks are demonstrated through human judgement scores.

1 Introduction

Inducing a particular style in generated text is a promising development which can lead to producing acceptable responses in dialogue generation, image captioning and artificial chat bot systems. In unsupervised text generation, estimating the distribution of real text from a corpus is a challenging task. Recent approaches using adversarial training have addressed this issue by trying to overcome the exposure bias that models trained for maximum likelihood suffer from. This work proposes an approach for text generation using a Generative Adversarial Network (GAN) with Skip-Thought vectors (STGAN). GANs (Goodfellow et al., 2014) are a class of neural networks

that explicitly train a generator to produce high-quality samples by pitting the generator against an adversarial discriminative model. GANs output differentiable values and the task of discrete text generation is challenging because of the non-differentiable nature of generating discrete symbols. Hence, in the present work, the GANs are trained with sentence embedding vectors as a differentiable input. The sentence embeddings are produced using Skip-Thought (Kiros et al., 2015), a neural network model for learning fixed length representations of sentences.

People's way of expression and communication intention is more diverse across utterances than the vocabulary. To imitate this, the proposed STGAN architecture models the variability at the utterance level in a corpus rather than at word or character level. The effectiveness of this approach is evaluated on automated corpus-based metrics: BLEU-n (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE (Lin, 2004) using different embeddings: Average GloVe (Pennington et al., 2014), Vector Extrema GloVe (Pennington et al., 2014) and Skip-Thought (Kiros et al., 2015). We perform an empirical study with human judgements to assess both the quality and the style reproduction in the generated text.

2 Related Works

Deep neural network architectures have demonstrated strong results on natural language generation tasks such as dialogue response generation and machine translation. Early techniques for generating text conditioned on some input information were template or rule-based engines (McRoy et al., 2000), or probabilistic models such as n-gram. In the recent past, state-of-the-art results on these tasks have been achieved by recurrent (Press et al., 2017; Mikolov et al., 2010) and con-

volutional neural network models trained for likelihood maximization. Very recently, attempts have been made to generate text using purely generative adversarial training (Arjovsky et al., 2017).

Unsupervised 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., 2017). Recent text generation models have used a wide variety of GANs such as policy-gradient based sequence generation framework (Yu et al., 2016). Fedus et al. (2018) have used an actor-critic conditional GAN to fill in missing text conditioned on the surrounding text for natural language generation tasks. GANs have also been used for text style transfer by Yang et al. (2018) where language models act as the discriminator and by Chen et al. (2018) with the introduction of a new f-measure termed as feature-mover’s distance.

Using adversaries of word and character level embeddings for text generation has been explored by Rajeswar et al. (2017). Models trained using generative adversarial networks or variational autoencoders have been shown to learn representations of continuous structures by leveraging deep latent variables such as text embeddings (Zhao et al., 2017). This work explores injecting sentence embeddings produced using the Skip Thought architecture (Kiros et al., 2015) into GANs in different setups.

3 Skip-Thought Generative Adversarial Network

In literature corpora such as fantasy and science fiction novels, the vocabulary does not vary significantly across the authors, but the manner of expression does, which is intuitively best captured at the level of sentences than words. The approach, hence, that this work takes in generating sentences with the writing style of one author is to make the adversarial model approximate the distribution of all sentences (rather than words or characters) in a latent space using skip-thought architecture. Previous attempts on text generation have used the character and word-level embeddings instead with GANs (Rajeswar et al., 2017).

This section introduces Skip-Thought Generative Adversarial Network with a background on neural network models that it is based on. The Skip-Thought model (Kiros et al., 2015) produces embedding vectors for sentences present in train-

ing 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 the course of training and then 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. This combination is identical to the RNN encoder-decoder of Cho et al. (2014) used in neural machine translation.

Skip-Thought Architecture

For a given sentence tuple (s_{i-1}, s_i, s_{i+1}) , let w_i^t denote the t -th word in sentence s_i and x_i^t denote its word embedding. The model is described as:

Encoder. Encoded vectors for a sentence s_i with N words w^i, w^{i+1}, \dots, w^n are computed by iterating over the following sequence of equations:

$$\begin{aligned} \mathbf{r}^t &= \sigma(\mathbf{W}_r \mathbf{x}^t + \mathbf{U}_r \mathbf{h}^{t-1}) \\ \mathbf{z}^t &= \sigma(\mathbf{W}_z \mathbf{x}^t + \mathbf{U}_z \mathbf{h}^{t-1}) \\ \mathbf{h}^t &= \tanh(\mathbf{W} \mathbf{x}^t + \mathbf{U}(\mathbf{r}^t \odot \mathbf{h}^{t-1})) \\ \mathbf{h}^t &= (\mathbf{1} - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \mathbf{h}^t \end{aligned}$$

where h_i^t is a hidden state at each time step and interpreted as a sequence of words w_i^1, \dots, w_i^n , \mathbf{h}^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,

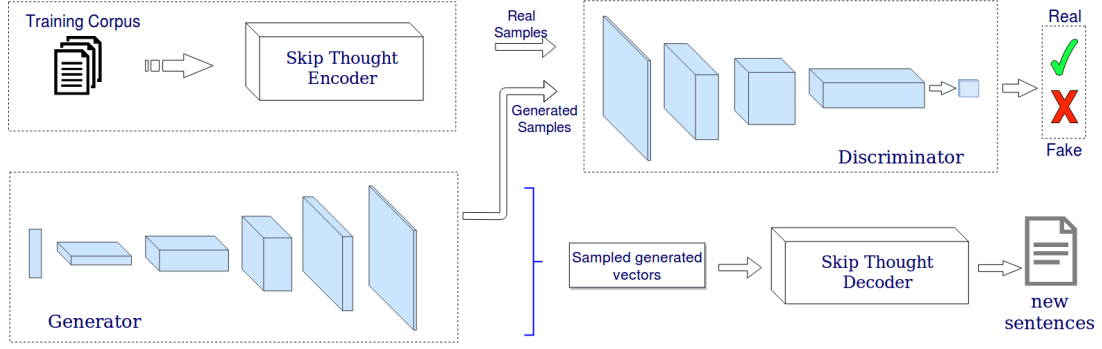


Figure 1: Skip-Thought Generative Adversarial Network model architecture

one each for sentences s_{i+1} and s_{i-1} . The following equations are iterated over for decoding:

$$\begin{aligned} \mathbf{r}^t &= \sigma(\mathbf{W}_r^d \mathbf{x}^{t-1} + \mathbf{U}_r^d \mathbf{h}^{t-1} + \mathbf{C}_r \mathbf{h}_i) \\ \mathbf{z}^t &= \sigma(\mathbf{W}_z^d \mathbf{x}^{t-1} + \mathbf{U}_z^d \mathbf{h}^{t-1} + \mathbf{C}_z \mathbf{h}_i) \\ \mathbf{h}^t &= \tanh(\mathbf{W}^d \mathbf{x}^{t-1} + \mathbf{U}^d (\mathbf{r}^t \odot \mathbf{h}^{t-1}) + \mathbf{C} \mathbf{h}_i) \\ \mathbf{h}_{i+1}^t &= (\mathbf{1} - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \mathbf{h}^t \end{aligned}$$

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:

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

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:

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

3.3 Model Architecture

The STGAN architecture (Figure 1) has two components: Skip Thought encoder-decoder and a generative adversarial network. The model uses a deep convolutional generative adversarial network, similar to the one used in DCGAN (Radford et al., 2015). During the training, 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 using 2400 GRU units (Chung et al., 2014) with a word vector dimensionality of 620. The encoder combines the sentence embeddings to produce 4800-dimensional combine-skip vectors with the first 2400 dimensions being uni-skip model and the last 2400 bi-skip model. This work uses the 4800-dimensional vectors as they have been found to be the best performing in experiments¹. For training of the STGAN, the Skip-Thought encoder produces sentence embedding vectors which are labelled as real samples for GAN discriminator.

The decoder uses greedy decoding by taking the argmax over the softmax output distribution for a given time-step which also acts as input for next time-step. It reconstructs sentences conditioned on a sentence vector by randomly sampling

¹<https://github.com/ryankiros/skip-thoughts/>

from the predicted distributions with a preset beam width. A 620 dimensional RNN word embedding is used for the decoder with 1600 hidden GRU decoding units. All experiments are performed using the Adam optimizer (Kingma and Ba, 2014) with gradient clipping and a batch size of 16.

Each sentence is appended with a start token $\langle s \rangle$ and an end token $\langle /s \rangle$ before encoding. During the process of training generator network with these embeddings, some generated vectors are randomly sampled. The sampled vectors are decoded using pretrained Skip-Thought decoder to produce a probability distribution over the vocabulary in order to reconstruct sentences. The decoding is terminated when the stop token $\langle /s \rangle$ is encountered during reconstruction.

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., 2016) have been suggested for making the training more stable. Training the Skip-Thought GAN often results in mode dropping (Arjovsky and Bottou, 2017; Srivastava et al., 2017) 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 a GAN. Gulrajani et al. (2017) have incorporated a gradient penalty regularizer term 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 certain attributes to generate real valued data (Mirza and Osindero, 2014; Radford et al., 2015). In this experiment,

both the generator and discriminator are conditioned on the Skip-Thought encoded vectors.

The data used for this setup consists of 250,000 sentences chosen from the BookCorpus dataset (Zhu et al., 2015) with a training/test/validation split of 5/1/1. All the sentences belong to one series of fantasy novels by a particular author of English language. This selection implies that the author’s word choice, sentence structure, figurative language, and sentence arrangement are consistent and well-represented across the dataset. Conditioning on this high-level outline gives more robustness to the model in terms of generated samples.

The decoded sentences form the evaluation set for measuring performance of different models under corpus level BLEU-n (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE(Lin, 2004) metrics. Table 1 compares these results for the proposed STGAN against standard LSTM and Attention Bidirectional LSTM models in two settings - using Skip Thought vectors and using tied GloVe (Pennington et al., 2014) embeddings. For this experiment, GloVe Average is obtained by averaging GloVe embeddings of all the words composing a given sentences while GloVe Extreme is computed by taking the most extreme value for each dimension across embeddings of all the words. All the three models used in the experiment: LSTM, Attention BiLSTM and Wasserstein GAN take the above three embeddings as input in separate runs and output vectors which are decoded to reconstruct sentences using the corresponding embedding’s decoder.

Table 2 shows improvements in metric scores when using Wasserstein distance and gradient penalty regularizer as discussed in section 3.4. WGAN-GP gives the strongest across-the-board performance in both the GloVe and Skip-Thought settings, so we use this as the basis for the rest of our experiments. The METEOR scores are reportedly better for other models because though it does not rely on embeddings but it includes notions of synonymy and paraphrasing to compute alignment between hypothesis and reference sentences (Sharma et al., 2017).

4.2 Language Generation.

Language generation is performed on a dataset comprising simple English sentences referred to

MODEL	EMBEDDING	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE
LSTM	GLOVE AVERAGE	0.874	0.792	0.621	0.582	0.681	0.692
	GLOVE EXTREME	0.874	0.791	0.616	0.580	0.677	0.685
	SKIP THOUGHT	0.885	0.807	0.633	0.585	0.683	0.692
ATTENTION BiLSTM	GLOVE AVERAGE	0.904	0.836	0.645	0.583	0.695	0.698
	GLOVE EXTREME	0.886	0.827	0.643	0.581	0.689	0.696
	SKIP THOUGHT	0.900	0.827	0.651	0.589	0.692	0.715
WGAN -GP	GLOVE AVERAGE	0.879	0.807	0.668	0.585	0.694	0.702
	GLOVE EXTREME	0.853	0.799	0.666	0.579	0.689	0.697
	SKIP THOUGHT	0.903	0.836	0.682	0.594	0.692	0.731

Table 1: Evaluation of models on word-overlap based automated metrics when trained with different embeddings. Skip-Thought gives better results than GloVe for BLEU-n and ROUGE metrics, while the METEOR scores are comparable to that when using averaged GloVe embedding with Attention BiLSTM generator.

MODEL	BLEU-2		BLEU-3		METEOR		ROUGE	
	GLOVE	ST	GLOVE	ST	GLOVE	ST	GLOVE	ST
GAN	0.710	0.745	0.593	0.607	0.667	0.670	0.654	0.649
WGAN	0.786	0.833	0.645	0.669	0.681	0.681	0.681	0.675
WGAN-GP	0.807	0.836	0.668	0.682	0.694	0.692	0.702	0.731

Table 2: BLEU-2, BLEU-3 METEOR and ROUGE metric scores across GAN models with different f-measures. *GloVe*: GloVe Average, *ST*: Skip-Thought, *WGAN*: Wasserstein GAN, *GP*: Gradient Penalty

as CMU-SE² (Rajeswar et al., 2017). The CMU-SE dataset consists of 44,016 sentences with a vocabulary of 3,122 words. The vectors are extracted in batches of same-lengthed sentences for encoding. The samples represent how mode collapse is manifested when using least-squares distance (Mao et al., 2016) f-measure without minibatch discrimination. Table 3(a) contains sentences generated from a vanilla STGAN which mode collapse is observed, while 3(b) contains examples wherein it is not observed when using minibatch discrimination. Table 3(c) shows generated samples from STGAN when using Wasserstein distance f-measure as WGAN (Arjovsky et al., 2017) and 3(d) contains samples when using a gradient penalty regularizer term as WGAN-GP (Gulrajani et al., 2017). The two models generate longer human-like sentences and over a more diverse vocabulary.

4.3 Human Scores and Correlations

The performance of this approach to generate new sentences has been evaluated in reproducing writ-

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

ing style of a particular author. The participant group consisted of 14 individuals, who were familiar with writing style of the said author by having read all but a few of the literary works of the author. The setup prevents them from being certain whether a sentence in question has or has not appeared in any work of the author that they have already read. To form the evaluation set of sen-

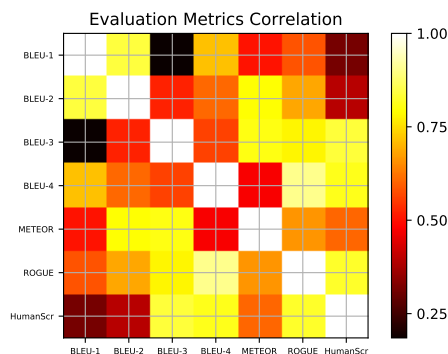


Figure 2: Pearson’s correlation coefficient between automated computed metrics and human scores. Human scores correlate well with BLEU-3 and ROUGE scores.

Model	Generated Samples
a. GAN (Mode collapse)	<ol style="list-style-type: none"> 1. it ? 2. it ? 3. it ? 4. it ? how would it ? 5. it ? how would it ?
b. GAN (minibatch)	<ol style="list-style-type: none"> 1. it a bottle ? 2. a glass bottle ? 3. a glass bottle it ? 4. it my hand a bottle ? 5. the phone my hand it
c. Skip Thought WGAN	<ol style="list-style-type: none"> 1. we have new year s holidays, always. 2. here you can nt see your suitcase , 3. please show me how much is a transfer? 4. i had a police take watch out of my wallet . 5. here i collect my telephone card and telephone number
d. Skip Thought WGAN-GP	<ol style="list-style-type: none"> 1. my passport and a letter card with my card , please 2. here on my telephone, mr. kimuras registration cards address. 3. i can nt see some shopping happened . 4. get him my camera found a person s my watch . 5. delta airlines flight six zero two from six p.m. to miami, please?

Table 3: Sentences sampled from STGAN when training on CMU-SE Dataset; mode collapse is overcome by using minibatch discrimination. Sample quality in terms of length and diversity further improved by using Wasserstein distance f-measure with gradient penalty regularizer. **WGAN**: Wasserstein GAN, **GP**: Gradient Penalty

	Real	Fake	% real	% fake
Real	30	51	37.04%	62.96%
Fake	48	75	39.02%	60.98%

Table 4: Weighted human scores for sentences. $|rating - 3|$ is weight given to each sentence’s rating. 39.02% of the generated samples were marked as real.

tences, the generated samples were mixed with real sentences from the author’s writing. 10 sentences from this mixed pool were chosen at random to be presented to each person. The participants were asked to mark on a scale of 1 to 5 if they thought that a sentence seemed to belong to the author’s works or was generated from a model, with 1 being certainly from the author and 5 being certainly from a model. Table 4 shows the weighted scores computed as $|rating - 3|$ to account for the degree of uncertainty addressed by a participant when rating. The models performs well with 39.02% of the generated samples being marked as written by the author while a greater 62.96% of the actual sentences from author’s writing being marked as fake generated

ones. Figure 2 compiles Pearson’s correlation coefficients between the obtained human scores and Skip-Thought GAN scores.

5 Conclusion

This work presents a simple and effective model for text generation based on adversarial training using sentence embeddings. It shows how the use of sentence-level embeddings allows modelling the way of expression of an author in generated text in a better way than when using word-level embeddings. A performance comparison across several metrics is made between different GAN architectures with improved training stability and attention augmented LSTM models. Finally, it discusses how the automated corpus-based evaluations correlate with human judgements. In future, this work aims to be applied for synthesizing images from text, exploring complementary architectures to projects like neural-storyteller³ where skip-thought embeddings are already used to perform image captioning with story-style transfer.

³<https://github.com/ryankiros/neural-storyteller>

Acknowledgements

The author would like to thank Aruna Malapati for providing insights and access to an Nvidia Titan X GPU for the experiments; and Pranesh Bhargava, Greg Durrett and Yash Raj Jain for providing helpful feedback. The author also acknowledges the support of Microsoft Research India Travel Grant.

References

- M. Arjovsky and L. Bottou. 2017. [Towards Principled Methods for Training Generative Adversarial Networks](#). *ArXiv e-prints*.
- M. Arjovsky, S. Chintala, and L. Bottou. 2017. [Wasserstein GAN](#). *ArXiv e-prints*.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72. Association for Computational Linguistics.
- Liquan Chen, Shuyang Dai, Chenyang Tao, Dinghan Shen, Zhe Gan, Haichao Zhang, Yizhe Zhang, and Lawrence Carin. 2018. [Adversarial text generation via feature-mover’s distance](#). *CoRR*, abs/1809.06297.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. 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.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- W. Fedus, I. Goodfellow, and A. M. Dai. 2018. [MaskGAN: Better Text Generation via Filling in the_____](#). *ArXiv e-prints*.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.
- Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30*, pages 5767–5777.
- Diederik P. Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#). *CoRR*, abs/1412.6980.
- Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.
- Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. 2016. Multi-class generative adversarial networks with the L2 loss function. *CoRR*.
- Susan W McRoy, Songsak Channarukul, and Syed S Ali. 2000. Yag: A template-based generator for real-time systems. In *Proceedings of the first international conference on Natural language generation-Volume 14*, pages 264–267. Association for Computational Linguistics.
- Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. 2016. Unrolled generative adversarial networks. *CoRR*, abs/1611.02163.
- Tomas Mikolov, Martin Karafiat, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*.
- Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *CoRR*, abs/1411.1784.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL ’02*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Ofir Press, Amir Bar, Ben Bogin, Jonathan Berant, and Lior Wolf. 2017. Language generation with recurrent generative adversarial networks without pre-training. *CoRR*, abs/1706.01399.
- Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434.
- Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Joseph Pal, and Aaron C. Courville. 2017. Adversarial generation of natural language. *CoRR*, abs/1705.10929.

- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*.
- Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. [Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation](#). *CoRR*, abs/1706.09799.
- A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton. 2017. [VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning](#). *ArXiv e-prints*.
- Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. 2018. [Unsupervised text style transfer using language models as discriminators](#). *CoRR*, abs/1805.11749.
- Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2017. Recent trends in deep learning based natural language processing. *CoRR*, abs/1708.02709.
- Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2016. Seqgan: Sequence generative adversarial nets with policy gradient. *CoRR*, abs/1609.05473.
- J. Zhao, Y. Kim, K. Zhang, A. M. Rush, and Y. LeCun. 2017. [Adversarially Regularized Autoencoders](#). *ArXiv e-prints*.
- Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.