Creative GANs for generating poems, lyrics, and metaphors

Asir Saeed
MLT Labs
asir@mltokyo.ai

Suzana Ilić
University of Innsbruck, MLT Labs
suzana@mltokyo.ai

Eva Zangerle
University of Innsbruck
eva.zangerle@uibk.ac.at

Abstract

Generative models for text have substantially contributed to tasks like machine translation and language modeling, using maximum likelihood optimization (MLE). However, for creative text generation, where multiple outputs are possible and originality and uniqueness are encouraged, MLE falls short. Methods optimized for MLE lead to outputs that can be generic, repetitive and incoherent. In this work, we use a Generative Adversarial Network framework to alleviate this problem. We evaluate our framework on poetry, lyrics and metaphor datasets, each with widely different characteristics, and report better performance of our objective function over other generative models.

1 Introduction and related work

Language models can be optimized to recognize syntax and semantics with great accuracy [1]. However, the output generated can be repetitive and generic leading to monotonous or uninteresting responses (e.g "I don't know") regardless of the input [2]. While application of attention [3, 4] and advanced decoding mechanisms like beam search and variation sampling [5] have shown improvements, it does not solve the underlying problem. In creative text generation, the objective is not strongly bound to the ground truth—instead the objective is to generate diverse, unique or original samples. We attempt to do this through a discriminator which can give feedback to the generative model through a cost function that encourages sampling of creative tokens. The contributions of this paper are in the usage of a GAN framework to generate creative pieces of writing. Our experiments suggest that generative text models, while very good at encapsulating semantic, syntactic and domain information, perform better with external feedback from a discriminator for fine-tuning objectiveless decoding tasks like that of creative text. We show this by evaluating our model on three very different creative datasets containing poetry, metaphors and lyrics.

Previous work on handling the shortcomings of MLE include length-normalizing sentence probability [6], future cost estimation [7], diversity-boosting objective function [8, 2] or Suppenalizing repeating tokens [9]. When it comes to poetry generation using generative text models, Zhang and Lapata [10], Yi et al. [11] and Wang et al. [12] use language modeling to generate Chinese poems. However, none of these methods provide feedback on the quality of the generated sample and hence, do not address the qualitative objective required for creative decoding. For the task of text generation, MaskGAN [13] uses a Reinforcement Learning signal from the discriminator, FMD-GAN [14] uses an optimal transport mechanism as an objective function. GumbelGAN [15] uses Gumbel-Softmax distribution that replaces the non-differentiable sample from a categorical distribution with a differentiable sample to propagate stronger gradients. Li et al. [2] use a discriminator for a diversity promoting objective. Yu et al. [16] use SeqGAN to generate poetry and comment on the performance of SeqGAN over MLE in human evaluations, encouraging our study of GANs for creative text generation. However, these studies do not focus solely on creative text.

2 GANs for creative text generation

Using GANs, we can train generative models in a two-player game setting between a discriminator and a generator, where the discriminator (a binary classifier) learns to distinguish between real and fake data samples and the generator tries to fool the discriminator by generating authentic and high quality output [17]. GANs have shown to be successful in image generation tasks [18] and recently, some progress has been observed in text generation [14, 13, 16]. Our generator is a language model trained using backpropagation through time [19]. During the pre-training phase we optimize for MLE and during the GAN training phase, we optimize on the creativity reward from the discriminator. The discriminator's encoder has the same architecture as the generator encoder module with the addition of a pooled decoder layer. The decoder contains 2 $[Dense, BatchNormalization, ReLU]$ blocks and 1 $[Dense, BatchNormalization, Sigmoid]$ block. The discriminator decoder takes the hidden state at the last time step of a sequence concatenated with both the max-pooled and mean-pooled representation of the hidden states [20] and outputs a number in the range $[0, 1]$. The difficulty of using GANs in text generation comes from the discrete nature of text, making the model non-differentiable hence, we update parameters for the generator model with policy gradients as described in Yu [16].

We utilize AWD-LSTM [21] and TransformerXL [22] based language models. For model hyperparameters please to refer to Supplementary Section, Table 2. We use the Adam optimizer [23] with $\beta 1 = 0.7$ and $\beta 2 = 0.8$ similar to [20] and use a batch size of 50. Other practices for LM training were the same as [22] and [21] for Transformer-XL and AWD-LSTM respectively. We refer to our proposed GAN as Creative-GAN and compare it to a baseline (a language model equivalent to our pre-trained generator) and a GumbelGAN model [15] across all proposed datasets. We use three creative English datasets with distinct linguistic characteristics: (1) A corpus of 740 classical and contemporary English poems, (2) a corpus of 14950 metaphor sentences retrieved from a metaphor database website [1] and (3) a corpus of 1500 song lyrics ranging across genres. The mix of linguistic styles within this corpus offers the potential for interesting variation during the generation phase. We use the same pre-processing as in earlier work [20, 24]. We randomly sample 10% of our data for test set and another 10% for our validation set.

We first pre-train our generator on the Gutenberg dataset [25] for 20 epochs and then fine-tune [20] them to our target datasets with a language modeling objective. The discriminator's encoder is initialized to the same weights as our fine-tuned language model. Once we have our fine-tuned encoders for each target dataset, we train in an adversarial manner. The discriminator objective here is to score the quality of the creative text. The discriminator is trained for 3 iterations for every iteration of the generator, a practice seen in previous work [26]. Creative-GAN relies on using the reward from the discriminator [13, 16] for backpropagation. We follow a similar training procedure for GumbelGAN. Outputs are generated through sampling over a multinomial distribution for all methods, instead of $argmax$ on the log-likelihood probabilities, as sampling has shown to produce better output quality [5]. Please refer to Supplementary Section Table 3 for training parameters of each dataset and Table 2 for hyperparameters of each encoder. We pick these values after experimentation with our validation set. Training and output generation code can be found online[2].

3 Evaluation and conclusion

Evaluating creative generation tasks is both critical and complex [27]. Along the lines of previous research on evaluating text generation tasks [27], we report the perplexity scores of our test set on the evaluated models in the Supplementary Section, Table 1 Our model shows improvements over compared methods. Common computational methods like BLEU [28] and perplexity are at best a heuristic and not strong indicators of good performance in text generation models [29]. Particularly, since these scores use target sequences as a reference, it has the same pitfalls as relying on MLE. The advantages in this approach lie in the discriminator's ability to influence the generator to explore other possibilities. Sample outputs for our model can be found in the Supplementary Section Table 4 and on our website [3].

---

[1]http://metaphors.iath.virginia.edu/

[2]https://github.com/Machine-Learning-Tokyo/Poetry-GAN

[3]https://www.ai-fragments.com

References

[1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

[2] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

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

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[5] Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[6] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[7] Allen Schmaltz, Alexander M. Rush, and Stuart M. Shieber. Word Ordering Without Syntax. In *Empirical Methods in Natural Language Processing*, number 1, pages 2319–2324, 2016.

[8] Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models. In *Empirical Methods in Natural Language Processing*, pages 2210–2219, 2017.

[9] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304, 2017.

[10] Xingxing Zhang and Mirella Lapata. Chinese Poetry Generation with Recurrent Neural Networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680, 2014.

[11] Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. Generating chinese classical poems with RNN encoder-decoder. *CoRR*, abs/1604.01537, 2016.

[12] Qixin Wang, Tianyi Luo, Dong Wang, and Chao Xing. Chinese song iambics generation with neural attention-based model. *CoRR*, abs/1604.06274, 2016.

[13] William Fedus, Ian Goodfellow, and Andrew M. Dai. MaskGAN: Better Text Generation via Filling in the_____. In *International Conference on Learning Representations*, 2018.

[14] Liqun Chen, Shuyang Dai, Chenyang Tao, Dinghan Shen, Zhe Gan, Haichao Zhang, Yizhe Zhang, and Lawrence Carin. Adversarial Text Generation via Feature-Mover's Distance. In *32nd Conference on Neural Information Processing Systems*, 2018.

[15] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[16] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *Association for the Advancement of Artificial Intelligence*, page 2852–2858, 2017.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[18] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *Neural Information Processing Systems*, page 1486–1494, 2015.

[19] Michael C Mozer. A focused backpropagation algorithm for temporal. *Backpropagation: Theory, architectures, and applications*, 137, 1995.

[20] Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018.

[21] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*, 2017.

[22] Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[23] Diederik P Kingma and Jimmy Lei Ba. Adam: Amethod for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations*, 2014.

[24] Jeremy Howard et al. fastai. `https://github.com/fastai/fastai`, 2018.

[25] Shibamouli Lahiri. Complexity of Word Collocation Networks: A Preliminary Structural Analysis. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–105, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[26] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv e-prints*, page arXiv:1701.07875, January 2017.

[27] Peter Potash, Alexey Romanov, and Anna Rumshisky. Evaluating creative language generation: The case of rap lyric ghostwriting. In *Proceedings of the Second Workshop on Stylistic Variation*, pages 29–38, 2018.

[28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. BLEU : a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, 2002.

[29] Lucas Theis, Aaron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016.

4   Supplementary Material

In this section, we report our results on computational metrics, hyperparameters, outputs from our model and their training configurations. Table 1 shows the the perplexity score of the evaluated models, Table 2 shows hyperparameters for each encoding method and Table 3 shows our training parameters. In Table 3, the values for Gutenberg dataset in columns, GumbelGAN and Creative-GAN are empty as we only pre-train our LMs with the Gutenberg dataset. Table 4 shows some curated outputs from our model.

	AWD-LSTM			Transformer-XL		
	Poetry	Metaphor	Lyrics	Poetry	Metaphor	Lyrics
LM	50.73	63.59	20.08	47.46	**62.76**	16.11
GumbelGAN	55.03	68.72	22.19	46.27	63.43	12.58
Creative-GAN	**49.40**	**51.84**	**17.11**	**42.45**	65.35	**9.02**

Table 1: Perplexity Scores, **bold** denotes best performance

Model	W. Emb. Size	Layers	Hidden	Backprop though time [19]
AWD-LSTM	400	3	1150	70
Transformer-XL	410	12	1024	150

Table 2: Encoder Hyperparameters

	LM		GumbelGAN		Creative-GAN	
	Epochs	LR	Epochs	LR	Epochs	LR
Poems	8	$3e-3$	10	$3e-4$	10	$3e-4$
Metaphors	8	$3e-4$	10	$3e-4$	10	$3e-4$
Lyrics	15	$3e-4$	12	$3e-4$	12	$3e-4$
Gutenberg	20	$3e-3$	–	–	–	–

Table 3: Training Parameters

Output Samples			
Frail, frail, fiery phantom, Death's wan, ghostly form soared Into the ether; threatening To burn her forever, and die	Time, the years, the adventures, The weeks, the years; The delicate sex, Hold thy tongue, the winds, The happy gardens, The county walks!	My Imagination is a Storehouse of all that is within	Yet he at times Would think about his village, and would wonder If the children and the rocks were still the same But as he grew older.
'Stranger, we are always the braver sort,' Said the Butcher, 'and your name's Dan Snark!'	Your help has given countless people around the world a voice - the one thing they need most	I couldn't sleep, and it thought ALL Was mean, wrong, and stupid	The views Of the ideal Conqueror are sage and gay;
Oh! Horrid Night! Melody release thee from my aching Heart, And Fate copied on my Mind, the wandering Misfortunes?	Home, home! Home, sweet home! Home, sweet! Home! Home! Home! Home!	Nothing else could ever move me.	I couldn't sleep, and it thought ALL Was mean, wrong and stupid.
Ours learned the wisdom of Ancient king. I remember the wild useless war	Accommodate all I ask, And all I want, Is to speak to myself.	Oh I like it, I like it Oh I like it, I like it Oh I like it, I like it, I like it	So many dreams go DAY AND NIGHT So many changes in my life So many right and so many bruises
For here I am, Weeping alone in your heart, Unnoticed by all I know	The analogy upon this hypothesis is something of the importance [and importance]. But the notion that[the imagination] itself is a storehouse of causes...	Over the green horizon, evanescently, I crossed the dusk. Trembling, dazed With yearning...	Her baby skin her old black dress Her hair is a blacker, Her heart is a deeper sin and she...
All the pleasing illusions, which made power gentle, and obedience apprehensions the taste of our love to return to the dictates of mankind; eternal thought, like the chameleon, governs the whole mind at the virtual stroke...	You see it in miles, in amply offered flight, And how it takes the form of a fire: But it always carries more than wood, Into business, illusions,happiness.	The world was a thing confining, And there was nothing like the Rain— So that I wept for cursing And I wept for the right of washing my grain.	All night long I am pursued by terror Because we know How we are safely held Under chains of prison skies
I said that the prayers in the Common Prayer Book were such as were planted in the Mind when they don't desire to see. They found that the most disrupted genes were those for growth factors, proteins that act like a kind of brain fertilizer a body is a forest in which you re-ascend	She lay under the midnight moon Her restless body stirring Until the magic morning hour Like poison it succumbs her	No angel of mercy Is gonna hear my call(oh no) No na na And you can hear my voice	On the meaning of tomorrow In a voice I hear warning white as thoughts coming to me Come, sing me back a little louder girl Early in the morning to fall to sleep

Table 4: Curated fragment samples from Creative GANs on the test dataset