Adversarial Conversational Shaping : Intelligent Chatting Agent

Piotr Tarasiewicz - UCL - piotr.tarasiewicz.20@ucl.ac.uk
Sultan Kenjeyev - UCL - sultan.kenjeyev.20@ucl.ac.uk
Ilana Sebag - UCL - ilana.sebag.20@ucl.ac.uk
Shehab Alshehabi - UCL - shehab.alshehabi.20@ucl.ac.uk

Abstract

The recent emergence of Deep Learning methods enabled the achievement of state-of-the art results in several domains including Natural Language Processing (NLP). Nonetheless, most of the time, text generator and chat-bots can be inefficient, boring and misunderstand human-like dialogue. Indeed, the current robocall systems employed by various businesses have many flaws : not being able to actually help you is just one of them. To tackle these issues, we propose to compare different models with different training and aim at presenting an intelligent conversational agent through adversarial conversational shaping. In this work, we compare Generative Adversarial Network with Policy Gradient and Generative Adversarial Network with Reward for Every Generation Step (REGS) (Li et al., 2017b). The latter model is able to assign rewards to both partially and fully generated text sequences. We will discuss different training details : Seq2Seq (Sutskever et al., 2014) and Transformers (Vaswani et al., 2017) in a Reinforcement Learning framework.

1 Introduction

Nowadays, sequential data is omnipresent and crucial : audio, video, text, time series are accessible to everyone. This allows the development of multiple state-of-the-art Machine Learning models in many fields such as Machine Vision (MV) and Natural Language Processing (NLP). Over the last few years, text generation became the flagship NLP research topic following the development and improvement of models algorithms : Recurrent Neural Networks (RNNs), Long Short Term Memory (LSTM), Gated Recurrent Units (GRUs) and BiDirectional RNNs (BRNNs). Text generation lays the foundation for many applications, especially, Open Dialogue Generation (Ritter et al., 2011) (Shen et al., 2018), Text Summarization (An et al., 2021) and Data Augmentation in NLP (Feng et al., 2021). Most of the time, these systems are built upon and end-to-end model : sequence-to-sequence model (Seq2Seq) (Sutskever et al., 2014) that aims at encoding an input text sequence into a mathematical vector and then decoding the vector into a target text sequence. In our work, we will compare the Seq2Seq paradigm with the brand-new T5 transformers (Raffel et al., 2020). The latter is a pre-trained encoder-decoder model in which each task is converted into a human-language-like task.

The objective of a Dialogue System is to generate coherent and meaningful text responses given a dialogue input. The standard training method for such neural language models usually uses a Maximum Likelihood Estimator (MLE) with the corresponding objective function derived from the Kullback-Leibler (KL) divergence between the empirical probability distribution representing the data and the parametric probability distribution output by the model (Labeau and Cohen, 2019). Despite the efficiency of this estimator, it leads to conventional answers and lack of originality. Indeed, the MLE estimates the parameters of a probability distribution by maximizing the corresponding likelihood function so that under the assumed statistical model the observed data is more probable. Thus, it encourages the model to generates high-frequency words such as 'are', 'the', 'and', 'is' and it is harder to produce rare words and interesting answers.

To tackle the above explained issue we are implement a GAN with Policy gradient and a GAN with Reward for Every Generation Step (REGS) using pre-training only in a Reinforcement Learning framework.

2 Related Work

Dialogue Generation Open-Domain Dialogue Generation is an increasingly prominent compo-

nent of Natural Language Processing (NLP). Indeed, nowadays, it constitutes the foundation of most NLP research. The techniques used in NLP for text generation have evolved alongside the progress in deep learning : Variational Auto-Encoders are now widely used, for instance for text summarizing (Miao and Blunsom, 2016) and dialogue modelling (Wen et al., 2017). Nonetheless Variational Auto-Encoder models have limitations due to ignorance of latent variables. Multiple researchers, such as (Vinyals and Le, 2015), (Serban et al., 2016), (Luan et al., 2016), used the Seq2Seq method in order to build end-to-end conversational systems. Furthermore, Generative Adversarial Network (GANs) have also been the flagship improvement in NLP (Haidar and Rezagholizadeh, 2019), (Rajeswar et al., 2017). Finally, the combination of Reinforcement Learning (RL) and Natural Language Processing (NLP) is becoming omnipresent in the field (Ramamurthy et al., 2020), (Luketina et al., 2019). Indeed, the reward system allows one to classify words efficiently, translate text accurately or generate dialogues. Our work uses a Reinforcement Learning framework for REGS and Policy Gradient with Seq2Seq and transformers T5.

Generative Adversarial Networks (GAN) are a Deep Learning technique that makes use of two neural networks : a generative model G and a discriminative model D that are trained simultaneously. G captures the distribution of the target data whilst D contributes to the training of G by classifying the generated data by G as real or machine-generated data. Adversarial Networks first appeared in 2014 (Goodfellow et al., 2014) as a pair of simple neural networks. This technique enables to generate new data with the same statistical properties as the input data used for the training set. Since then, Generative Adversarial Networks have been widely used in different fields especially in Computer Vision (Radford et al., 2016), (Chen et al., 2016) and Natural Language Processing (NLP) (Glover, 2016), (Li et al., 2017a). GANs have also been combined with Reinforcement Learning framework in order to improve multiple generation tasks such as speech language generation with Policy Gradient Reinforcement Learning by back-propagating the error from the discriminator (Yu et al., 2017).

Transfer Learning is widely used in the field of Machine Learning to reuse, transfer and leverage knowledge from a model. It is a popular approach in Deep learning where pre-trained models are used as the starting point on computer vision (Li et al., 2020) and Natural Language Processing tasks (Ruder et al., 2019). In our work, we use the transformers model T5 that was first presented by (Raffel et al., 2020) in order to convert all text-based language problems into a text-to-text format.

3 Generative Adversarial Networks (GANs) for Conversational Shaping

Given a dialogue history x , we aim to generate responses y according to a policy defined by an encoder-decoder model. We defined two different models : a GAN with REGS and a GAN with Policy Gradient. Then, we will compare the usage of Seq2Seq and T5 Transformers on these models.

3.1 GAN with Policy Gradient

The GAN is composed of a generative model G and a discriminative model D. G defines a policy that generates the response y by computing a probability vector of each token in the target sequence using a softmax function whilst D has the role of a classifier. In this case, we consider a binary discriminator that uses a sequence of pair of input and response dialogue $\{x, y\}$ and classify each input as either human generated or machine generated. Based on the work of (Li et al., 2015), we encode the input sequences into a vector of probabilities with the help of a Hierarchical Neural Auto-encoder. When the input is classify as human-generated, the corresponding assigned score is denoted by $Q_+(\{x, y\})$ and when the input is classify as machine-generated, the corresponding assigned score is denoted by $Q_-(\{x, y\})$.

For the training, we use Policy Gradient algorithm : It is a Reinforcement Learning technique that aims at optimizing the parametrized policy with respect to the long-term cumulative reward by gradient descent. This training method encourages the model to generate human-like responses y which are generated by sampling directly from the policy and used to update the discriminator. In this framework, the assigned score is used as reward for the generator. Thus, the objective is to maximize the expected reward. We do so using the REINFORCE algorithm from (Williams, 1992) as

in (Li et al., 2017a).

$$J(\theta) = E_{y \sim p(y|x)}(Q_+(\{x, y\}) \mid \theta)$$

And, its gradient can be estimated as follows :

$$\nabla J(\theta) \approx [Q_+(\{x, y\}) - b(\{x, y\})] \nabla \log \pi(y \mid x)$$

$$= [Q_+(\{x, y\}) - b(\{x, y\})] \nabla \sum_t \log p(y_t \mid x, y_{1:t-1})$$

In the above equations, π represents the probability of the generated responses y and b is the baseline used to regulate the variance of the estimate to make it efficient (low variance, unbiased and consistent).

3.2 GAN with Reward for Every Generation Step (REGS)

In the previous section, we saw that the Generative Adversarial Network (GAN) with Policy Gradient model presents some flaws : a unique reward is assigned to each token of the human-generated response whilst we would expect different rewards. Also, in REGS, the discriminative model aims at assigning rewards to both fully and partially generated text sequences whilst the neural network uses the mean squared loss between the machine-generated text and real text rewards. This proves the importance of computing the reward at each intermediary step of the generation. (Li et al., 2017b) propose two strategies to compute such rewards : Monte Carlo Search and training a discriminator that is able to assign rewards to partially and fully generated sequences. In this section, we focus on reproducing the latter strategy from (Li et al., 2017b)'s paper.

Like in (Rajeswar et al., 2017), we denote the generated sequences $\{y_{1:t}\}_{t=1}^{N_Y}$ and separate them into positive and negative sequences to denote the partially generated sequences, namely $\{y^+_{1:t}\}_{t=1}^{N_{Y+}}$ and $\{y^-_{1:t}\}_{t=1}^{N_{Y-}}$. Then, we randomly sample one example from each sequence and use it to update the discriminator. For each partially generated sequence Y_t, the discriminator assigns a score $Q_+(x, Y_t)$ that will classify the sequence. The corresponding baseline value denoted $b(x, Y_t)$ helps regulate the variance (Ranzato et al., 2016).

Hence, the generator is updated according to :

$$\nabla J(\theta) \approx \sum_t (Q_+(x, Y_t) - b(x, Y_t))$$

$$\nabla \log p(y_t \mid x, Y_{1:t-1})$$

3.3 Training details

Given the dialogue history, we start by pre-training the generator by predicting target sequences. Then we train both Seq2Seq with attention and transformers T5 models for each neural network.

We experimented several procedures in order to assess efficiency of the models :

- We trained the networks with and without teacher forcing.
- We trained the networks with and without layer freezing. When we applied layer freezing, we froze all layers of the model except the language model head and last decoder block of the transformer.
- We tried different range of learning rates : $1e^{-3}$, $3e^{-4}$ and $5e^{-5}$. After several trials, we observed that the most suited learning rate was $1e^{-3}$.

3.3.1 Sequence-to-sequence (Seq2Seq) with attention

In this section, we evaluate the pretrained and REGS models with Seq2seq with attention. We used the code from (Li et al., 2017a). The Seq2Seq framework relies on the encoder-decoder paradigm : it consists on encoding the source sequences and decoding the target sequence. The attention mechanism enforces the model to pay more attention on specific parts of the source sequence when decoding. That is, the encoder does not have to encode the entire input sequence into a vector and we are not relying only on the hidden vector anymore (Bahdanau et al., 2016).

During training, at each time step, the decoder will generate a probabilistic vector $p_i \in R$ that contains the probabilities of each token at each relevant time step. Then, given the input sequence x_i, we can compute the probability of some target sequence y_i : $P(y_1, ..., y_m) = \Pi_{i=1}^m p_i[y_i]$; where $p_i[y_i]$ means that we extract the y_ith entry of the vector p_i from the ith decoding step. We aim at generating human-like text, that is, maximizing the probability of the target sequence. This is the same as minimizing the standard cross entropy between the target distribution and the actual output :

$$-\log P(y_1, ..., y_m) = -\log \Pi_{i=1}^m p_i[y_i]$$

$$= -\sum_{i=1}^m \log p_i[y_i]$$

3.3.2 Transformers T5

We also built our neural networks using a Transformers T5 for comparison purpose. T5 is an encoder-decoder model was first presented in (Raffel et al., 2020)'s work. It is pre-trained on a multi-task mixture of unsupervised and supervised tasks and treat every text processing problem as a "text-to-text" problem. The advantage of the text-to-text format is that we can apply the same model, loss function, hyper-parameters, training procedure and decoding procedure on both the input and the output.

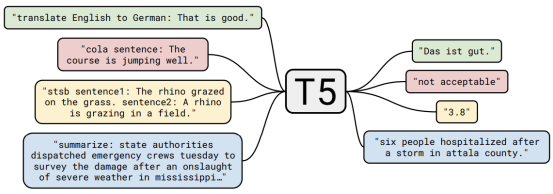

Figure 1: Explanatory diagram of the T5 framework : "Text-to-Text Transfer Transformer" , image from (Raffel et al., 2020)

We used the language model for both the generator and the discriminator. Indeed, as suggested in (Raffel et al., 2020), using the language model instead of a binary classifier would improve the accuracy of the predictions in this case.

3.3.3 Teacher Forcing

Teacher Forcing is widely used in the field of Deep Learning Language Models to quickly and efficiently train RNNs that use the ground truth from a prior time step as input. This method consists on supplying observed sequence values as inputs during training and using the network's own one-step ahead predictions to do multi-step sampling as explained in (Lamb et al., 2016). For instance, we would give both human and machine generated responses to the generator for model updates and arbitrarily assign the value of 1 a s reward to each human generated response. That is, the 'teacher' is forcing the model to learn what is a human-like generated response. The idea of this method is to enforce the model the regularize itself when it deviates from the training dataset.

In our experiments, we tried to implement our models with and without Teacher Forcing. It presents some advantages and disadvantages. This method will enforce a faster converging training as the hidden states of the model are not updated with wrong prediction sequences anymore, nonetheless, the issue of Exposure Bias occurs : (Schmidt, 2019). When no ground-truth is available, there is a train-test discrepancy that might lead to inefficiency of the model.

4 Experimental Results and Discussions

We evaluate the above detailed methods on the Daily Dialogue dataset (Li et al., 2017c) and asses their efficiency using adversarial evaluation. We trained the T5 discriminator from scratch and raised the accuracy of predicting real and fake generated responses on balanced data. We also did the same procedure manually by denoting ourselves whether the dialogue and the corresponding generated answer made sense or not. We obtained the following results :

	T5 PT	T5 REGS	T5 PG
Dist-1	0.2	0.085	0.079
Dist-2	0.47	0.177	0.162
Bleu-1 (1e-3)	98.88	58.26	56.84
Bleu-2	55.2	26.66	24.96
Bleu-3	21.01	8.57	6.84
Bleu-4	14.21	6.1	1.07
Adversarial Accuracy	68%	70%	72%
Human Accuracy(avg)	64%	73%	68%

Table 1: Final Results for T5 (PG stands for Policy Gradient and PT stands for pretrained)

	S2S PT	S2S REGS
Dist-1	0.078	0.081
Dist-2	0.551	0.502
Bleu-1 (1e-3)	30.06	41.74
Bleu-2	15.8	23.3
Bleu-3	10.16	4.22
Bleu-4	5.11	1.32
Adversarial Accuracy	85%	75%
Human Accuracy(avg)	81%	80%

Table 2: Final Results for Seq2Seq (PT stands for pretrained)

5 Conclusion

In this work, we draw intuitions from the work of (Li et al., 2017a), (Xu et al., 2018) and (Zhu et al., 2020). We propose an adversarial training approach for response generation. We built two Generative Adversarial Networks using the brand new text-to-text transformers T5 and compared it

with a Seq2Seq implementation and a pretrained model. For the evaluation, we used BLUE and DIST. We choose cast the models in a Reinforcement Learning framework and deal with assigned scores as rewards for the classifiers. This allowed the generator to generate human-like dialogue responses.

From the Tables 1 and 2 displayed above, we can conclude that out of all the presented models, the pretrained T5 model is the one that performs the best with the lowest adversarial accuracy of 68%. Furthermore, we notice that all T5-based networks perform better than the Seq2Seq-based networks with a lowest adversarial accuracy of 80% obtained with the Seq2Seq REGS model. We can assume that T5 performs better Seq2Seq as it is a powerful model, thus, the discriminator might easily cheat by finding hacks. Finally, we can observe that the GAN seems to contribute more to the efficiency of the model when paired with seq2seq rather than with T5.

We also manually assessed the models by acting like the discriminator ourselves and classifying the generated response as real or fake given an input dialogue. We obtained the same ranking : pretrained T5 seems to perform the best and all T5-based models perform better than Seq2Seq-based models.

6 Future work

We identified two main axes of expansions for this project. On the first hand, implementing a Diversity-Promoting GAN would allows to assess and compare the efficiencies of the models. On the other hand, exploring and applying counterfactual reasoning to these models could show great improvement.

Diversity-Promoting Generative Adversarial Network (DP-GAN) : (Xu et al., 2018) implemented a DP-GAN in a Reinforcement Learning framework. This model is contains a language model based discriminator D trained over real and generated text, in opposition to our binary classifier, and assigns low reward to repetitive text and high reward for novel and rare text, to encourage the generator to produce more diverse text.The output of the discriminator is used as reward for the generator.

Counterfactual Reasoning is a psychology concept that describes human behaviour able to learn from previous experiences and create alternative solutions. It is a probabilistic answer to the question "what would have happened if ". In the case of our work, counterfactual reasoning allows a bot to more accurately answer a question and participate to a discussion. In the area of NLP, Deep Learning and reinforcement Learning, counterfactual reasoning is used for different purposes. Most commonly, it is used for data augmentation purposes (Kaushik et al., 2020), (Zmigrod et al., 2019). But also, in order to explore alternative policies that an agent could have been taken (Zhu et al., 2020). But also, in the purpose of leveraging advantages of counterfactual reasoning for decision making in the reinforcement learning framework (Buesing et al., 2018). Or another field of application of counterfactual reasoning is Learning representations as in (Johansson et al., 2018). As extension of this work, applying a counterfactual inference system on the trained models should improve the diversity of the responses. (Zhu et al., 2020) implemented a CF framework on the REGS neural network.

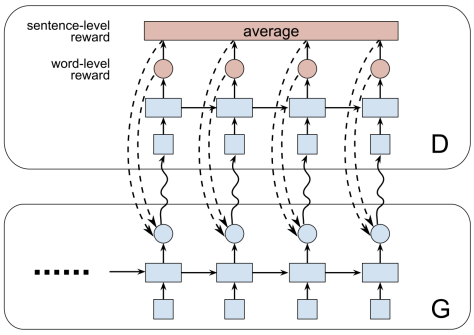

Figure 2: DP-GAN model , image from (Xu et al., 2018)

References

Chenxin An, Ming Zhong, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2021. Enhancing scientific papers summarization with citation graph. *CoRR*, abs/2104.03057.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Lars Buesing, Theophane Weber, Yori Zwols, Sébastien Racanière, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. 2018. Woulda,

coulda, shoulda: Counterfactually-guided policy search. *CoRR*, abs/1811.06272.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp.

John Glover. 2016. Modeling documents with generative adversarial networks.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks.

Md. Akmal Haidar and Mehdi Rezagholizadeh. 2019. Textkd-gan: Text generation using knowledgedistillation and generative adversarial networks.

Fredrik D. Johansson, Uri Shalit, and David Sontag. 2018. Learning representations for counterfactual inference.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data.

Matthieu Labeau and Shay B. Cohen. 2019. Experimenting with power divergences for language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4104–4114, Hong Kong, China. Association for Computational Linguistics.

Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks.

J. Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. *ArXiv*, abs/1701.06547.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents.

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017b. Adversarial learning for neural dialogue generation. *CoRR*, abs/1701.06547.

Xuhong Li, Yves Grandvalet, Franck Davoine, Jingchun Cheng, Yin Cui, Hang Zhang, Serge Belongie, Yi-Hsuan Tsai, and Ming-Hsuan Yang. 2020. Transfer learning in computer vision tasks: Remember where you come from. *Image and Vision Computing*, 93:103853.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017c. Dailydialog: A manually labelled multi-turn dialogue dataset.

Yi Luan, Yangfeng Ji, and Mari Ostendorf. 2016. Lstm based conversation models.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. 2019. A survey of reinforcement learning informed by natural language.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 319–328, Austin, Texas. Association for Computational Linguistics.

Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. 2017. Adversarial generation of natural language.

Rajkumar Ramamurthy, Rafet Sifa, and Christian Bauckhage. 2020. Nlpgym – a toolkit for evaluating rl agents on natural language processing tasks.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 583–593, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Florian Schmidt. 2019. Generalization in generation: A closer look at exposure bias. In *NGT@EMNLP-IJCNLP*, pages 157–167.

Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. 2016. Generative deep neural networks for dialogue: A short review.

Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg.

2018. Improving variational encoder-decoders in dialogue generation.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. 2017. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3732–3741. PMLR.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, pages 229–256.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018. Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3940–3949, Brussels, Belgium. Association for Computational Linguistics.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient.

Qingfu Zhu, Weinan Zhang, Ting Liu, and William Yang Wang. 2020. Counterfactual off-policy training for neural response generation. *CoRR*, abs/2004.14507.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.