A Review of Generative Adversarial Networks in Text Generation

Jaden Cohen

—---

Abstract

This paper presents a high-level exploration of Generative Adversarial Networks (GANs) and their potential role in the field of text generation, as well as their advancement in the past few years. I start by discussing the intricacies of natural language processing (NLP), specifically text generation, and the various challenges the field faces. I then dive into a detailed examination of various GAN models, each defined by its unique architecture and approach to overcoming the hurdles in text generation. I provide in-depth analyses of these key models, examining their strengths, limitations, and the specific text generation challenges they address. Furthermore, the paper identifies crucial issues in current GAN techniques, such as training instability and lack of output diversity. In response, I propose potential paths for future research, including the exploration of more compact and efficient GAN models. My conclusion highlights the significant potential of GANs in revolutionizing text generation, emphasizing their role in advancing AI's creative capabilities in language. This research not only serves as a valuable resource for those interested in the technical aspects of GANs but also acts as a gateway for future innovations in the rapidly evolving landscape of AI-driven text generation.

—---

1.1. Introduction

In recent years, AI has risen to the forefront of human technology as a powerful and innovative technology. Its versatility and variety make it applicable to various fields, from healthcare to finance to security. However, while those are powerful uses of AI, the focus of this paper will be on text generation. Now more than ever, with powerful models like GPT-4 available to the public, AI is being increasingly used by the public for more creative purposes, such as art generation, music generation, and, of specific interest to this paper, text generation. Writing underlies our society, putting the abstract ideas in our heads into concrete representations. The ability to take this innately human ability and train AI to utilize it has such significant potential, that it has an entire field of study dedicated to it, natural language processing, also known as

NLP. NLP applies a wide variety of different techniques to process, understand, utilize, and most crucially, generate text. There exist a variety of models used in text generation but one of the most interesting models in the field is known as the Generative Adversarial Networks (GAN), boasting an innovative approach to problems like text generation. Work has been done in this field before, most notably a paper published in 2022, titled A Survey on Text Generation Using Generative Adversarial Networks [1]. However, in the time since the paper was published, significant strides have been made, all of which will be comprehensively covered in this paper. I start by introducing GANs as well as their various architectures and designs. Then I discuss the successes and failures found in applying GANs to the field of text generation. Finally, I identify shortcomings in the field and suggest solutions.

1.2. NLP and Challenges

NLP is one of the most challenging fields of artificial intelligence, mostly due to the sheer ambiguity of language. Unlike math, language follows no absolute rules, and its only uniting principle is the characteristic of being interpretable by human inference, inference that AI models lack [2]. In language, phrases can be removed and added at will, having little effect on the ultimate meaning of the sentence. Add in additional complexities such as irony, humor, idiomatic expressions, sarcasm, multilingualism, and various other quirks of language, and the difficulty of natural language processing becomes apparent [3]. Of course, to combat the difficulty of this field numerous techniques have been created, including sentiment analysis, named entity recognition, keyword extraction, and others [4]. However, while these techniques aid greatly in the comprehension and synthesis of text, they unfortunately only help text generation, the focus of this paper, indirectly at best. Text generation is among the most complex fields of NLP, and faces a diverse set of issues, such as being forced to deal with the sheer flexibility of human language. While models simply focused on understanding text can simply extract keywords and ignore articles or semantically insignificant words, in text generation, those words are the ones that make the sentences legible. Furthermore, taking the sentence beyond merely legible and into the realm of varied and well-written is an entirely different challenge. Finally, a crucial deficiency underlies the entire field: insufficient data. Text generation models need notable amounts of high-quality, varied text to produce good results, but various roadblocks like cost restrictions and ethical considerations contribute to the shortage that bogs down the field. This lack of data leads to lackluster models and overfitting, transforming the result into something mundane and uninteresting [5]. There have been many attempts to overcome this challenge of lacking data, including various data augmentation techniques such as synonym replacement, and random swapping, inserting, and deleting [6]. However, these techniques cannot completely erase the need for high-quality training data, but instead merely ease the burden. To resolve this issue, an architecture capable of producing good results off of a limited amount of data must be developed. In recent years a promising new technique has emerged, one that shows the potential to deal with many of the issues facing text generation today, a Generative Adversarial Network, or a GAN.

2.1 GANs

The base technique of the Generative Adversarial Network (GAN) is reliant upon neural networks, a mathematical and computational structure designed to imitate human neurons. They have the ability to learn complex patterns and form the foundation for the subfield of machine learning that has come to be known as deep learning. GANs put two different deep neural networks, the generator and the discriminator, against each other in a zero-sum game of sorts. See Figure 1.

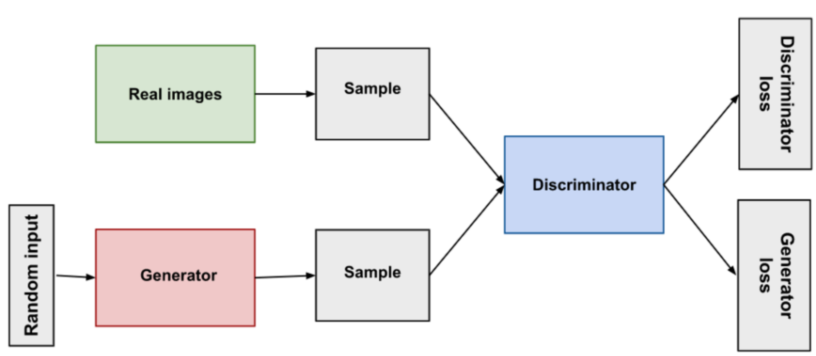

Figure 1: Traditional GAN architecture

The generator is in charge of producing fake data given a random noise vector, essentially a set of random values, and the discriminator, when fed a mixture of data produced by the generator and real data, is charged with assigning a probability that the given data is real [7].

The loss function below in Figure two is the overall cost function of GANs, the function that is optimized to produce the best results.

$$\min_G \max_D V(D, G)$$

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

Figure 2: Classic GAN loss function [8]

Z is the random latent vector given to the generator and X is the actual data from the given dataset. This function pits the two models that compose the GAN against each other, with the discriminator (D) trying to maximize the function while the generator (G) tries to minimize it. There are further variations of this function, but given that this paper is a high-level overview, I do not dive into the specifics.

The benefits of such an approach are apparent in the fact that since the generator never even sees the training data, it becomes much harder for bias to propagate, leading to much more varied and unique text, even with lacking training data. This inherent advantage allows it to meet the criteria necessary for solving the major lack of data and allows it to produce good results off a limited amount of training data. However, the approach is not without problems. One of the leading problems is that GANs were originally designed for generating images and other continuous values, or values that can take on any value within a given range. In contrast, language is discrete, meaning it can only take on a finite set of distinct values. This mismatch has caused significant problems in the generation of text via GANs. However, researchers from around the world have introduced many new techniques for overcoming this limitation, making GANs a viable and potent tool for text generation. For this paper, I review four key models that have been developed in the last few years to explore the current state of GANs in text generation and avenues for advancement.

2.2. ConcreteGAN

The ConcreteGAN uses an innovative mixture of both continuous and discrete learning methods [10]. In fact, this model delves so deeply into both sides that it can almost be considered two interconnected models. See Figure 3 for a visual representation.

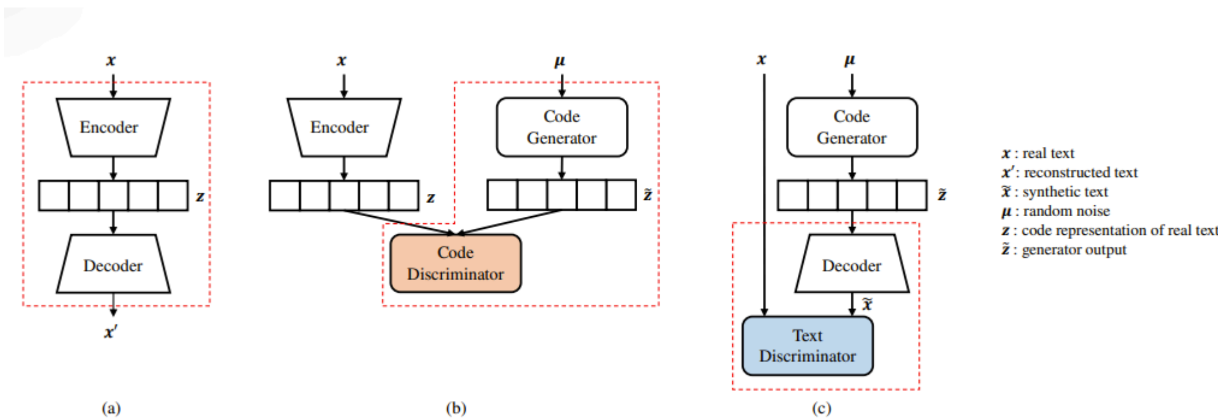

Figure 3: Concrete GAN [9]

Figure 3a is the autoencoder. Figure 3b is the continuous segment of the model. Figure 3c is the discrete segment.

First, it uses an autoencoder, pictured in Figure 3a, to transform the discrete text into a vector representation. Autoencoders are a special type of neural network that learns more compact ways to represent data by having the encoder section encode the data into a vector and having a matching decoder reconstruct it. This helps the model learn a more compact, and crucially, continuous representation of the discrete text. This continuous version of the data is then fed to the continuous GAN component of the model, through both a generator and a discriminator following the classical GAN architecture, pictured in Figure 3b. At the same time, the code generator is also running its outputs through the decoder, which once more transforms the data to a discrete representation, pictured in Figure 3c. That discrete data is run through another generator, this one using a policy gradient reinforcement learning method, which is a method that optimizes the generator based on the reward signal from the environment, which is necessarily higher the more correct the output. Here, the generator is treated as a stochastic policy, meaning a set of rules designed to achieve a certain goal, in this case generating good text, but with a bit of randomness incorporated to keep it from getting stuck at local maximums. It is optimized based on complete sequence evaluations, essentially feedback from the discriminator. Together, this model weaves a continuous and discrete section into a composite whole, neatly handling the largest problem of GAN text generation, namely its inability to handle discrete data, and creating a synergistic approach [9].

The inherent advantages of its approach are apparent in its high evaluations. It shows impressive performance in text generation, apparent across several datasets, including the COCO Image Caption, Stanford Natural Language Inference, and EMNLP 2017 WMT News datasets, and evaluation methods. Its Fréchet Distance (FD) scores, which measure the similarity between generated and real text, were particularly telling. The lower the FD score, the more similarity between the generated text and sample text. On the SNLI dataset, it achieved an FD score of 15.5, and on the EMNLP dataset, it scored 16.2, a marked improvement over it competition, the adversarially regularized autoencoder (ARAE), which scored a 24.7 on the SNLI dataset and a 18.9 on the EMNLP dataset. Furthermore, in human evaluations involving 100 randomly sampled sentences from each model, assessed by ten people on Amazon Mechanical Turk, ConcreteGAN outperformed other models. Specifically, on the EMNLP 2017 WMT News dataset, it received a human evaluation score of 3.337 (±0.946) highlighting its enhanced ability to generate realistic and contextually coherent text. Finally, the model was evaluated with BLEU scores, a number between 0 and 1 that represents the generated text's similarity to a high-quality reference. The higher the number after BLEU, also known as the n-gram value, the harder it is to get a higher score. It achieved BLEU scores of 0.871, 0.681, 0.466, and 0.311 for BLEU-2, BLEU-3, BLEU-4, and BLEU-5 respectively. These scores indicate the similarity of ConcreteGAN's output to reference texts. Additionally, backward BLEU (B-BLEU) scores, which assess the recall of generated text, were recorded as 0.817, 0.636, 0.446, and 0.301 for B-BLEU-2, B-BLEU-3, B-BLEU-4, and B-BLEU-5 respectively. These scores represent how well the generated text encompasses the diversity of the dataset. See Table 1 for ConcreteGAN evaluation metrics.

BLEU-2 (0-1)	BLEU-3 (0-1)	BLEU-4 (0-1)	BLEU-5 (0-1)	B-BLEU-2 (0-1)	B-BLEU-3 (0-1)	B-BLEU-4 (0-1)	B-BLEU-5 (0-1)	Human Eval (0-5)	FD scores
0.871	0.681	0.466	0.311	0.817	0.636	0.446	0.301	3.337	15.5-16.2

Table 1: ConcreteGAN evaluation metrics (explained above)

2.3. Adversarial Autoregressive Network (ARN)

Adversarial Autoregressive Network (ARN) was developed in response to the mode collapse problem in text generation, a problem where the generator gets stuck producing only a severely limited variety of data [10]. It combines autoregressive models, models that predict the future based on the past, like recurrent neural networks (RNNs) and their more advanced cousins, long short-term memory networks (LSTMs) with autoencoders, and a traditional GAN architecture to solve this problem. See Figure 4 for a visual representation.

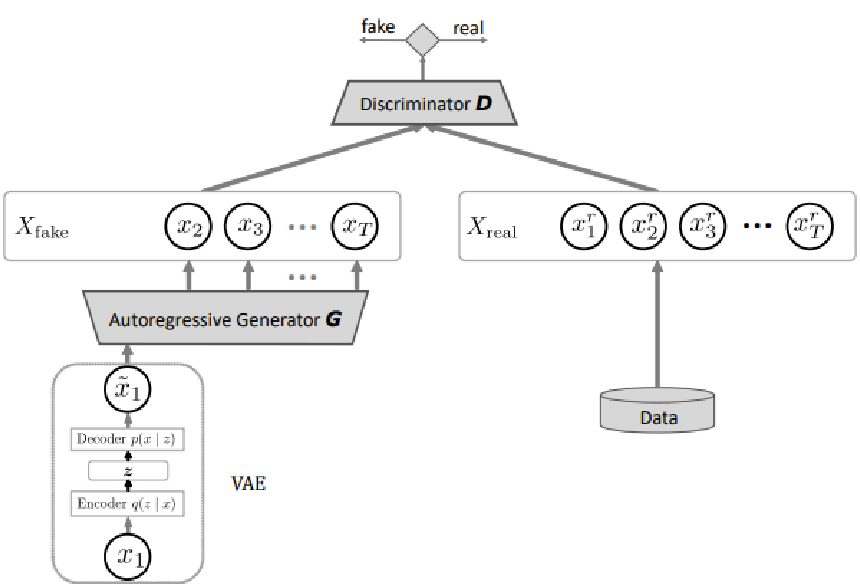

Figure 4: Adversarial Autoregressive Network [10]

Crucially, a variational autoencoder (VAE) is used, which for the purpose of this paper can just be thought of as an advanced autoencoder with a probabilistic instead of deterministic output. It transforms the initial input before feeding it to the generator, ensuring a compact continuous representation and addressing the mode collapse problem by providing a wide variety of inputs. The generator, which is autoregressive in nature (using RNNs or LSTMs), then takes over to produce the text sequence in a stepwise fashion. This setup is completed with a discriminator, as in standard GAN architectures, to form the complete model.

.

In evaluating the Adversarial Autoregressive Networks (ARN) model, specific metrics were used to assess the quality and diversity of generated text, most notably BLEU scores. ARN scored a 0.69 BLEU-2 score and a 0.3 BLEU-3 score when tested on the IMDB review dataset. Additionally, it was evaluated with FC (feature coverage) scores, a metric used to measure uniqueness by assessing how well it captures the variety of expressions found in the dataset. It scored 0.14 for FC-2 (2-gram) and 0.12 for FC-3 (3-gram), notably higher than its competitors. The diversity scores further highlight the ARN model's capability to generate a wide range of text, indicating its effectiveness in producing varied and innovative content. The model achieved diversity scores of 0.31 for Diversity-2 (2-gram diversity) and 0.64 for Diversity-3 (3-gram diversity), demonstrating its superior ability to create diverse and engaging text outputs compared to other models. This blend of high-quality and diverse text generation underscores the ARN model's advanced capabilities in handling complex natural language processing tasks. In summary, the ARN model demonstrated strong performance in both accuracy and diversity of generated text, balancing quality with uniqueness, and proving the architecture an effective one. See Table 2 for the scores.

BLEU-2 (0-1)	BLEU-3 (0-1)	FC-2 (0-1)	FC-3 (0-1)	Diversity-2 (0-1)	Diversity-3 (0-1)
0.69	0.3	0.14	0.12	0.31	0.64

Table 2: Evaluation metrics for ARN Adversarial Autoregressive Network

2.4. Feature Aware Conditional GAN (FA-GAN)

FA-GAN is a more recent model, developed in 2023, that was designed to address many of the issues typically faced by text-generation GANs, including mode collapse, training instability, lack of diversity, and controllability [11]. One key difference compared to most of the other models examined in this paper is that the FA-GAN can generate text from specific prompts, not just random noise. See Figure 5 for a diagram of its inner workings.

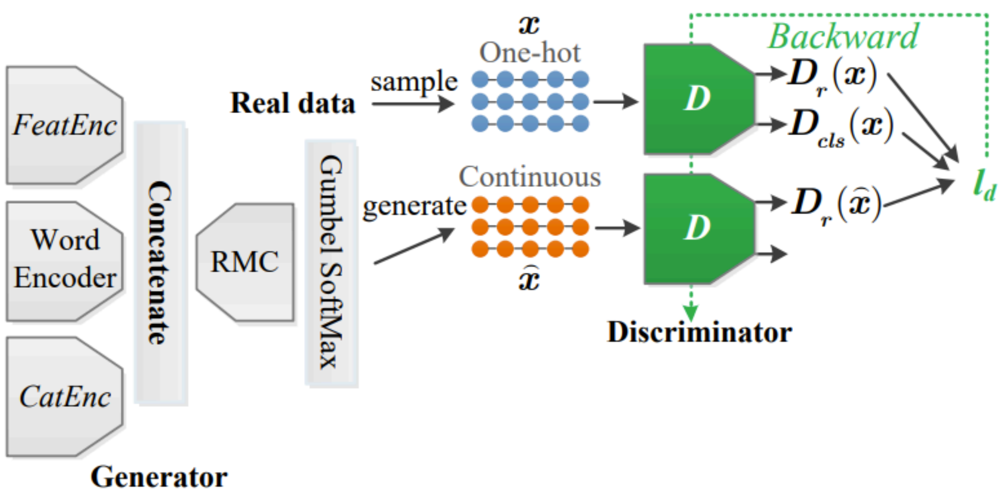

Figure 5: Feature Aware Conditional GAN [11]

The architecture works by passing the prompt separately through three encoders: a feature encoder, responsible for understanding context, a category encoder, responsible for embeddings of the specific category to be generated, and a word encoder, which stores strong word embeddings, essentially continuous vector representations of them, in a fixed table. These embeddings are then concatenated, or added together, to create a comprehensive representation which is then passed to the rational memory core (RMC), a module in the generator's decoder that enhances text generation by handling long-range dependencies through a self-attention mechanism across multiple memory slots. A differentiable Gumbel softmax function, a mathematical function that squeezes values between one and zero, is then used to make a discrete choice on the next word, essentially allowing the continuous output to be turned into a discrete choice while still allowing backpropagation, the method in which deep networks learn. From there, the generated text is fed into the discriminator, which not only differentiates between real and generated but also between categories—which allows backpropagation to not only ensure the text is realistic—but also that it is of the appropriate category.

FA-GAN's performance was evaluated on several text classification tasks. It demonstrated the highest accuracy across all datasets in comparison with other methods. Specifically, on the MR-20 dataset, it achieved a classification accuracy of 69.74%, which was comparable to advanced models like CBERT and GPT-2. On the Senti140-20 dataset, it surpassed other models like SSMBA, T5, SentiGAN, and CatGAN by more than 1.3%. In a low-data regime (MR-10-Low), FA-GAN improved accuracy by 2.58% over the non-augmented approach and performed significantly better than most other methods except for SentiGAN. Its BLEU scores likewise reflected its high performance, always being notably higher than other models. Finally, its negative log-likelihood diversity, a metric of variety, maintained levels nearly double that of its competitors, emphasizing the diversity and variance of its text. See Table 3 for FA-GAN evaluation metrics.

Classification Accuracy (0-100)	BLEU-2 Scores (0-1)	BLEU-3 Scores (0-1)	Negative Log-likelihood Diversity
62.32% - 88.32%	0.346 - 0.767	0.159 - 0.489	1.604 - 2.618

Table 3: Evaluation metrics for FA-GAN across the MR-10, MR-20, AM-30, USAir-20, Senti140-20, and MR-10- Low

2.5. Text generation with GANs using Feedback Score

Developed in 2023, the Feedback Score GAN (FC-GAN) centers around the concept of using feedback to enhance the quality of generated text, addressing many issues inherent to these models such as instability and uncontrollability [12]. It uses a largely traditional GAN with a Wasserstein cost function, an alternate cost function that mitigates many of the problems commonly present in GANs like vanishing gradients and mode collapse. However, the most important part of it is the incorporation of feedback, which it applies to the loss function. The feedback scores are designed to numerically assess the realism of the text and modify the generator loss accordingly. The more realistic the output the less the feedback adds to the loss. See Figure 6 for more information.

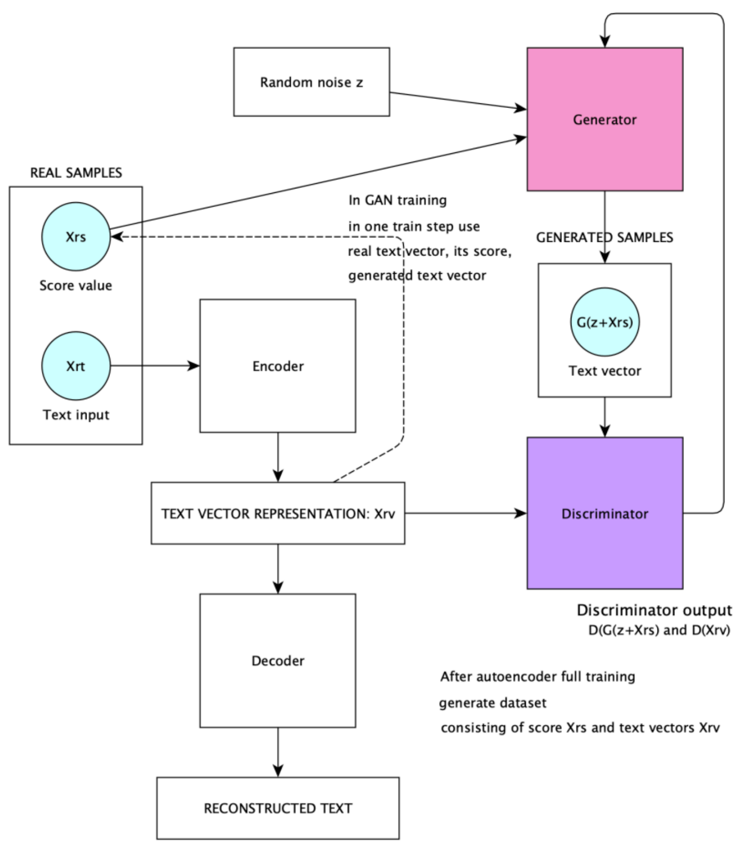

Figure 6: Feedback Score GAN [14]

z – generated noise input, Xrt – real text input, Xrv - real text vector processed by AE, Xrs – real feedback score value, gp – gradient penalty (from WGAN).

The model was evaluated using BLEU and BERT (F1) scores, comparing the generated text to real text. The BLEU scores, which measure the similarity to reference text, were notably low, ranging from 0.00308 to 0.03080 across various examples. This indicates a significant divergence of the generated text from the reference text. However, the BERT scores, assessing the fidelity and relevance, were much higher, with values ranging from 0.8087 to 0.8584. These scores imply that despite the low similarity, the generated text maintains a reasonable level of relevance and quality. This juxtaposition of low BLEU scores with higher BERT scores suggests that the model is adept at producing new and unique responses, while still keeping them coherent and understandable. See Table 4 for a complete picture of FC-GAN evaluation scores.

BLEU Scores	BERT Scores
0.00308 - 0.03080	0.8087 - 0.8584

Table 4: FC-GAN evaluation metrics

3.1. Analysis

Model	Basic Architecture	BLEU-2 Score	BLUE-3 Score	BERT Score (F1)	FD Score
FC-GAN	Traditional GAN with feedback	-	-	0.8087 to 0.8584	-
FA-GAN	Triple-encoder with RMC and Gumbel Softmax	0.346 to 0.767	0.159 to 0.489	-	-
ConcreteGAN	Continuous and discrete methods with policy gradient	0.729 to 0.871	0.528 to 0.681	-	SNLI: 15.5, EMNLP: 16.2
ARN	Autoregressive generator with transformers and autoencoders	0.6904 to 0.6923	0.3066 to 0.3070	-	-

Table 5: A comparison of all models discussed in this paper

Over the past few years, there have been tremendous advances in NLP and text generation, and GANs stand as promising models among the rest. Together, all four models presented in this paper represent the best of those text-generation GANs, presenting a diverse set of techniques and advancements. The sheer flexibility of GANs along with their inbuilt advantages make them an ideal place for these new advancements.

Concrete GAN not only built upon reinforcement techniques to navigate the discrete space but also merged this approach with a more traditional one involving autoencoders in the continuous space, weaving them together into a synergistic whole. As an all-encompassing approach, it addresses issues in both the discrete and continuous space and leverages each other's strength to bypass any obstacles they encounter. This two-sided approach gives it a good grasp on the nuances of text generation, making it an excellent option in tasks requiring complex language such as creative writing and conversation agents.

The Adversarial Autoregressive Network (ARN), on the other hand, makes use of autoregressive networks typically found in text generation and utilizes the GAN framework to bring even more out of them. Additionally, it makes use of an innovative technique with variational autoencoders (VAEs) to avoid mode collapse, neatly solving another of text generation with GAN’s greatest issues. The variation it generates makes it ideal for more flexible styles of writing, such as creative writing where a wide variety of stylistic expressions are needed.

The Feature Aware Conditional GAN (FA-GAN) is another recent innovation that steps away from being limited to random text generation and gains the capability to process prompts and output relevant text, a feature that was not explicitly present in the paper “A Survey on Text Generation Using Generative Adversarial Networks”[1], showing clear growth in the field since its authoring. It shows the potential for GANs and their flexibility in the array of diverse techniques it employs such as multiple different encoders and a relational memory core. The ability to process user input is crucial to a large language model and FA-GAN proves that the GAN architecture is extremely compatible with that task. It is an excellent choice for fields that require targeted generation, such as content creation and dialogue systems.

Finally, the Feedback Score GAN (FC-GAN) introduces another metric of improvement, that of integrating feedback to improve results, another new and novel concept showcasing the field's growth. The greatly enhanced quality that resulted from this is yet another example of the potential of GANs. It represents a step towards more interactive and responsive models, where feedback can be integrated directly into the model itself. It represents a notable advancement in explaining and controlling GANs.

Together, these models represent a set of techniques that more than encompasses all the necessary requirements to becoming a dominant model in the field of text generation.

3.2. Problems and Suggestions

One of the biggest problems inherent to the GAN architecture itself is mode collapse, where the generator ends up producing a very limited amount of outputs, training instability, and a lack of diverse outputs. While the techniques above have proven effective, they have not completely solved the inherent problems of the architecture. I recommend more research be put in the direction of refining the base models, potentially exploring alternative strategies, and integrating newer innovations. For instance, using adaptive training methods such as Adaptive Multi Adversarial training (AMAT), an alternate architecture that uses multiple discriminators to balance the training process, to enhance the diversity of generated text while retaining stability could prove a fruitful investigation. Additionally, I recommend that a comprehensive strategy is developed from a set of smaller strategies like mini-batch discrimination, a technique that distinguishes samples within a mini-batch for diversity, and instance noise, a technique that adds randomness to inputs, and integrated into the GAN architecture itself, potentially resolving the issues with minimal excess computational load. Finally, I believe it could be productive to look into alternate cost functions that heavily penalize lack of diversity, steering GANs away from problems like mode collapse.

Another problem is the difficulty in evaluating text. Many different studies and papers use different metrics and datasets to evaluate their models. While it's not as bad as it could be, the lack of uniformity makes it difficult to compare and contrast models. Furthermore, even such standardized metrics often fail to fully capture the intangible power of good writing, leaving some models with artificially boosted ratings and some with artificially lower ratings. I recommend that an effort be made to find a more universal evaluation metric, whether that be a completely new metric or a new procedure of combining multiple to produce a comprehensive picture of a model's performance. Furthermore, most of the commonly used evaluation metrics, like BLEU, tend to focus mostly on the syntactic side of things, leaving its semantics unevaluated. To this end, I additionally recommend further emphasis be placed on semantic methods of evaluation, like BERT scores, to fully capture the essence of what the models are outputting. Finally, there is potential for integrating human feedback into the training cycle, allowing the model to gradually converge at text more in touch with human sensibilities.

Next, long sentences are still a struggle for these models. They unfortunately tend to lose track of where they are the longer the sentence gets, leaving them with very little variety of sentence structure. Techniques like LSTMs are effective in combating this, as is the relational memory core in the FA-GAN, due to their ability to track and store context in specially designed short-term memory modules. Even though these methods are still very flawed, as evidenced by the continual disability to generate long sentences, the idea of utilizing external memory components to store context shows potential, and I recommend it be researched further. Furthermore, a more comprehensive integration of more self-updating attention mechanisms, mechanisms that decide which parts of a sentence are the most semantically meaningful, into the model should prove effective in helping maintain coherence for longer. Finally, If long sentence generation remains difficult, an alternate method could be simply stepping around the problem, using chunking and other similar methods to break sentences into more manageable chunks before being processed.

Finally, GANs tend to have low computational efficiency, purely due to their complexity and ever-increasing size. This will likely pose a major threat to their continued usage. To that end, I recommend that more efficient versions of current models are studied and implemented, for the purpose of making them easier to use and more practical. Additionally, I recommend research be done into how far one can reduce the precision of the neural networks' weights before its performance starts to deteriorate, potentially greatly reducing computational efficiency for very little in return. An extreme example of this could be potentially researching GANs with binary neural networks (BNNs), which use only binary weights and activations. Finally, there is potential in looking more at the hardware side of things, and investigating the interactions between these models and their hardware to maximize their efficiency.

3.3. Conclusion

In this paper, I have presented a thorough investigation of Generative Adversarial Networks (GANs) in the field of text generation, underscoring their significant contribution to the advancement of Natural Language Processing (NLP). My exploration has included innovative models like the Concrete GAN, which uniquely combines discrete and continuous data techniques for advanced language processing, and the Adversarial Autoregressive Network (ARN), specifically designed to address the challenge of mode collapse in GANs. The Feature Aware Conditional GAN (FA-GAN) stands out for its capability in prompt-based text generation, employing multiple encoders and a relational memory core, while the Feedback Score GAN (FC-GAN) introduces a novel integration of user feedback into the generative process, thereby enhancing the relevance and authenticity of the generated text. Despite these advancements, I recognize the persistence of challenges such as managing the intricacies of long and complex sentences, improving computational efficiency, and establishing universal metrics for evaluation. Moreover, the need to address training instability and adapt GANs to the inherently discrete nature of language remains a crucial area for further research. In conclusion, this paper highlights the substantial progress made with GANs in text generation, pointing towards a future where AI's ability to generate human-like text becomes increasingly refined and sophisticated, with ongoing research poised to enhance these capabilities even further.

Works Cited

[1] De Rosa, Gustavo, and João Papa. *A Survey on Text Generation Using Generative Adversarial Networks*. 2022.

[2] ABRO, Abdul Ahad, et al. "Natural Language Processing Challenges and Issues: A Literature Review." *Gazi University Journal of Science*, vol. 36, no. 4, 1 Dec. 2023, pp. 1522–1536, https://doi.org/10.35378/gujs.1032517. Accessed 8 Feb. 2024.

[3] James, Lora. "NLP Problems: 7 Challenges of Natural Language Processing." *MetaDialog*, 27 Apr. 2023, www.metadialog.com/blog/problems-in-nlp/#:~:text=Dealing%20with%20ambiguity%2C%20idiomatic%20phrases. Accessed 8 Feb. 2024.

[4] Wolff, Rachel. "Natural Language Processing (NLP): 7 Key Techniques." *MonkeyLearn Blog*, 19 Oct. 2021, monkeylearn.com/blog/natural-language-processing-techniques/.

[5] “Navigating through Text Generation Challenges and Solutions.” *AIContentfy*, 1 Sept. 2023, aicontentfy.com/en/blog/navigating-through-text-generation-challenges-and-solutions#:~:text=One%20of%20the%20main%20challenges. Accessed 8 Feb. 2024.

[6] “Text Data Augmentation in Natural Language Processing with Texattack.” *Analytics Vidhya*, 26 Feb. 2022, www.analyticsvidhya.com/blog/2022/02/text-data-augmentation-in-natural-language-processing-with-texattack/.

[7] Dey, Victor. “Beginner’s Guide to Generative Adversarial Networks (GANs).” *Analytics India Magazine*, 18 July 2021, analyticsindiamag.com/beginners-guide-to-generative-adversarial-networks-gans/. Accessed 8 Feb. 2024.

[8] Goodfellow, Ian, et al. *Generative Adversarial Nets*. 2014.

[9] Kim, Won, et al. *Collaborative Training of GANs in Continuous and Discrete Spaces for Text Generation.* 2016.

[10] Hossam, Mahmoud, et al. *Text Generation with Deep Variational GAN*.

[11] Li, Xinze, et al. “Feature-Aware Conditional GAN for Category Text Generation.” *Neurocomputing*, vol. 547, 1 Aug. 2023, pp. 126352–126352, https://doi.org/10.1016/j.neucom.2023.126352. Accessed 28 Oct. 2023.

[12] Kuznetsov, Dmitrii. “Text Generation with Gan Networks Using Feedback Score.” *Ssrn.com*, 10 May 2023, papers.ssrn.com/sol3/papers.cfm?abstract_id=4453193. Accessed 8 Feb. 2024.