

A Review of Generative Adversarial Networks in Text Generation

Jaden Cohen

Received August 31, 2024

Accepted November 30, 2024

Electronic access December 15, 2024

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. Specifically, I will analyze ConcreteGANs, Adversarial Autoregressive Networks (ARNs), Feature Aware Conditional GANs (FA-GANs), and Feedback Score GANs (FC-GANs). 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.

Keywords: Artificial intelligence, generative adversarial networks, text generation,

Introduction

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 architectures used in text generation but one of the most interesting ones 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, many 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.

A. 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 inter-

pretable by human inference, inference that AI models lack². 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 NLP becomes apparent³. Of course, to combat the difficulty of this field numerous techniques have been created, including sentiment analysis, named entity recognition, keyword extraction, and others⁴. 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 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

The ConcreteGAN uses an innovative mixture of both continuous and discrete learning methods¹¹. 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. First, it uses an autoencoder, pictured in

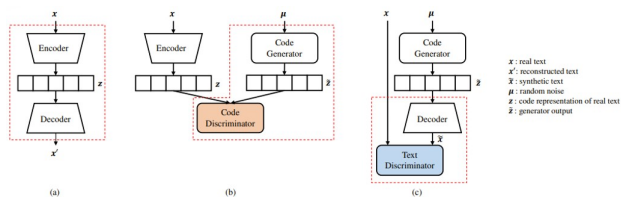


Fig. 3 Concrete GAN Architecture¹¹. Figure 3a is the autoencoder. Figure 3b is the continuous segment of the model. Figure 3c is the discrete segment.

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 learning¹² 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 the continuous and discrete sections into a composite whole, neatly handling the largest problem of GAN text generation, namely its inability to handle discrete data, and creating a synergistic approach¹¹.

The inherent advantages of its ConcreteGAN 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 text¹³. 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 its competition, the adversarially regularized autoencoder (ARAE), which scored a 24.7 on the SNLI dataset and an 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 reference¹⁴. 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.

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

Table 1 ConcreteGAN Model Evaluation Metrics for Text Generation

- Higher BLEU scores indicate syntactic similarity.
- Higher B-BLEU scores emphasize content recall.
- Lower FD scores reflect better quality and diversity by matching real data distributions.

• **B. 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 data¹⁵. 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. Crucially, a variational autoencoder (VAE)

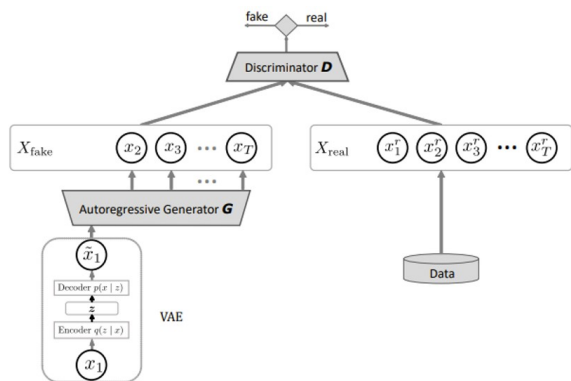


Fig. 4 Adversarial Autoregressive Network¹⁵. The VAE is a variational autoencoder

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 inputs¹⁶. 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.

Metric	Score	Metric	Score
BLEU-2 (0-1)	0.69	BLEU-3 (0-1)	0.3
BLEU-2 (0-1)	0.69	FC-2 (0-1)	0.14
FC-3 (0-1)	0.12	FC-2 (0-1)	0.14

Table 2 Adversarial Autoregressive Network Evaluation Metrics for Text Generation

- Higher FC scores indicate greater text diversity.
- Higher BLEU scores reflect closer syntactic alignment with the reference text.

C. 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 controllability¹⁷. 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.

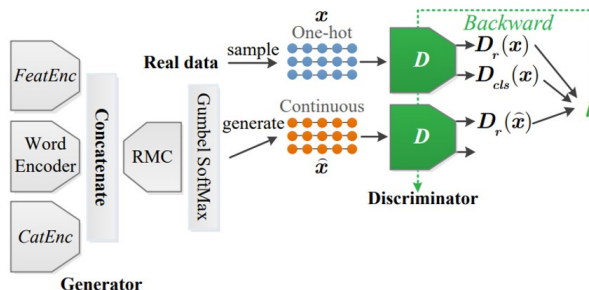


Fig. 5 Feature Aware Conditional GAN¹⁷

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 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 slots¹⁸. A differentiable Gumbel softmax function¹⁹, 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 variety²⁰, maintained levels nearly double that of its competitors, emphasizing the diversity and variance of its text. See Table 3 for FA-GAN evaluation metrics.

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

Table 3 FA-GAN Evaluation Metrics for Text Generation across the MR-10, MR-20, AM-30, USAir-20, Senti140-20, and MR-10-Low datasets

- Negative log-likelihood diversity indicates text variety.
- BLEU-2 and BLEU-3 scores measure syntactic similarity, with higher scores showing better alignment with the reference text.

D. 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 uncontrollability²¹. 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 collapse⁹. 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.

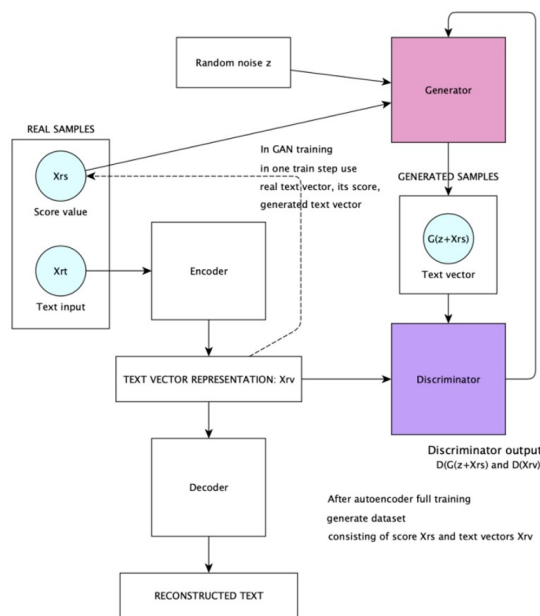


Fig. 6 Feedback Score GAN¹⁹ z – generated noise input, X_{rt} – real text input, X_{rv} - real text vector processed by AE, X_{rs} – real feedback score value, g_p – gradient penalty (from WGAN).

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 relevance²², 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.

Metric	Score Range
BLEU Scores	0.00308 - 0.03080
BERT Scores	0.8087 - 0.8584

Table 4 FC-GAN Evaluation Metrics for Text Generation

- BLEU scores measure syntactic similarity based on n-gram overlap.

-
- BERT scores assess semantic similarity using contextual embeddings, capturing the meaning of the text.

Analysis

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 builds upon reinforcement techniques to navigate the discrete space but also merges this approach with traditional methods involving autoencoders in the continuous space, weaving them into a synergistic whole. This comprehensive strategy addresses challenges in both discrete and continuous spaces, leveraging the strengths of each to bypass obstacles encountered in text generation. By balancing discrete and continuous representations, Concrete GAN demonstrates a robust understanding of the nuances of language modeling, making it a strong candidate for tasks requiring sophisticated language, such as creative writing and conversational agents. However, it's important to consider that the integration of these methods introduces complexity, potentially impacting the efficiency and scalability of the model. Further research could evaluate the trade-off between performance gains and computational cost. The Adversarial Autoregressive Network (ARN) uses autoregressive networks, a staple in text generation, within the GAN framework to enhance model performance. It also incorporates variational autoencoders (VAEs) to address mode collapse, which is a persistent challenge in GAN-based text generation. This innovative use of VAEs generates diverse and varied text, making ARN well-suited for creative applications where stylistic variation is crucial. Nevertheless, it's worth questioning whether the reliance on autoregressive techniques limits the model's speed and scalability in real-time applications. Additionally, the use of VAEs, while effective for increasing variation, could introduce biases in text generation based on the data distribution, a consideration that warrants deeper exploration. The Feature Aware Conditional GAN (FA-GAN) represents a significant innovation by extending beyond random text generation to prompt-based, context-aware generation. Unlike earlier GAN models, FA-GAN employs multiple encoders and a relational memory core, allowing it to process and respond to user input effectively. This advancement reflects growth in the field since the paper "A Survey on Text Generation Using Generative Adversarial Networks."¹ However, the model's reliance on structured prompts and the complexity of its relational memory mechanisms could introduce biases, especially in how it prioritizes different features in text generation. Addressing these biases and evaluating

their impact on the quality of generated text is essential for future research. Moreover, the model's performance should be scrutinized in terms of how well it generalizes to varied prompts outside its training data. Finally, the Feedback Score GAN (FC-GAN) takes a novel approach by integrating feedback into the training process to improve generation quality. This concept showcases significant potential for creating more interactive and responsive models. However, the reported BLEU scores, which range from 0.00308 to 0.03080, are notably lower than those of other models. These low scores suggest that while the feedback mechanism may be innovative, it may not be as effective at producing text that closely aligns with reference texts in terms of word choice and phrasing. This raises the question of whether the FC-GAN approach has a fundamental limitation or whether the BLEU metric is insufficient for capturing improvements made by the feedback integration. A more detailed analysis of these scores, alongside an exploration of alternative evaluation metrics that account for semantic meaning and contextual relevance, would provide a more balanced assessment. Additionally, potential biases in the evaluation process, such as the reliance on metrics like BLEU that favor surface-level similarity, should be considered, as they may not fully reflect the model's ability to generate diverse and meaningful content. 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.

E. 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 outputs²³. 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 investigation²⁴. 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. If such a strategy is integrated into the GAN architecture itself, it will potentially resolve the issues with minimal excess computational load²⁵. 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. Crucial to this effort is the ability to measure improvements.

Model	Basic Architecture	BLEU-2 Score	BLEU-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 All Models Evaluation Metrics for Text Generation

- BLEU scores measure syntactic similarity through n-gram overlap.
- BERT scores evaluate semantic similarity using contextual embeddings.
- FD (Fréchet Distance) assesses the overall quality and diversity by comparing data distributions.

Mode collapse detection metrics are essential to understanding and addressing the diversity issues inherent in GAN-generated text. Statistical measures, like entropy, can quantify the variety in generated text samples²⁶. Calculating entropy provides a clear, quantifiable signal of diversity: lower entropy values indicate reduced output variety, a hallmark of mode collapse, while higher values reflect a broader range of generated phrases. By applying these metrics, researchers can track improvements in diversity when implementing techniques like those discussed above. 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 significant potential for integrating human feedback into the training cycle, a strategy that could lead to models generating text that better aligns with human expectations and preferences²⁷. By incorporating human evaluations of output quality—focusing on aspects like fluency, coherence, relevance, and emotional impact—GAN models can receive more nuanced guidance than traditional auto-

generated metrics can provide. This feedback loop allows the model to iteratively learn and adjust, refining its understanding of what constitutes high-quality text. Over time, this process enables the model to produce content that feels more natural, engaging, and semantically meaningful, gradually converging on outputs that resonate with human sensibilities and practical applications, such as creative writing, customer service interactions, or content generation. Furthermore, this approach could also help address challenges like bias and lack of diversity in generated text by incorporating diverse perspectives in the feedback mechanism. Next, long sentences are still a struggle for these models. They 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 self-updating attention mechanisms²⁸ could significantly enhance the model's ability to generate coherent long sentences. These mechanisms dynamically determine which parts of a sentence are the most semantically meaningful, allowing the model to allocate focus more intelligently. By continuously adjusting the attention weights as the model processes each word, self-updating attention ensures that critical contextual relationships are preserved, even over extended sequences. This approach helps the model retain important information from earlier in the sentence while still responding to new input, thereby reducing the risk of losing track of the overall structure or meaning. 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 processed²⁹. 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 loss in accuracy. An extreme example of this could be potentially researching GANs with binary neural networks (BNNs), which use only binary weights and activations³⁰. 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.

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 avoiding mode collapse, 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.

GLOSSARY

- **Natural Language Processing (NLP):** A field of artificial intelligence focused on enabling computers to understand, interpret, and generate human language. It combines techniques from linguistics, computer science, and machine learning to process and analyze natural language data for tasks like translation, text generation, and sentiment analysis.
- **Policy Gradient Reinforcement Learning:** A method used in reinforcement learning where the model optimizes a policy directly using the gradients of expected rewards, effective for handling both discrete and continuous actions.
- **Autoencoders:** Neural networks that learn efficient data representations by compressing input data into a lower-dimensional space and then reconstructing it.
- **Variational Autoencoder (VAE):** A probabilistic version of an autoencoder that outputs a distribution, enabling more diverse and varied text generation and addressing mode collapse.
- **Relational Memory Core (RMC):** A memory module that enhances text generation by using self-attention to capture and maintain context over long sequences.
- **Fréchet Distance (FD):** A metric used to evaluate the similarity between the distributions of generated and real text, with lower values indicating better quality and diversity.
- **Gumbel Softmax:** A continuous approximation to categorical sampling, allowing discrete choices in text generation models while preserving the ability to use gradient-based optimization.
- **BERT (Bidirectional Encoder Representations from Transformers) Scores:** Metrics for evaluating semantic similarity, measuring how well the generated text maintains the intended meaning compared to the reference text.
- **BLEU (Bilingual Evaluation Understudy) Scores:** Metrics that assess the syntactic similarity of generated text to reference text based on n-gram overlap, with higher scores indicating better matches.
- **B-BLEU (Backward BLEU):** A variant of BLEU that focuses on recall, evaluating how well the generated text covers the content.
- **Mode Collapse:** A phenomenon where the GAN generator produces limited, repetitive outputs, failing to generate diverse and varied text.

- **Adaptive Multi-Adversarial Training (AMAT):** A training method that uses multiple discriminators to stabilize the GAN training process and improve text diversity.
- **Feature Coverage (FC) Scores:** Metrics that measure the diversity of generated text by assessing how well the model captures a range of features present in the training data.
- **Negative Log-Likelihood (NLL) Diversity:** A measure of the variety of generated text, with higher values indicating a broader range of expressions.
- **Self-Attention Mechanism:** A method that allows models to weigh different parts of a sentence based on their importance, improving coherence and understanding of long text sequences.
- **Recurrent Neural Networks (RNNs):** Neural networks that process sequences of data, maintaining context through hidden states, commonly used for text generation.
- **Long Short-Term Memory (LSTM):** A type of RNN designed to retain information over long sequences, addressing the vanishing gradient problem and enhancing model memory.
- **Continuous vs. Discrete Space:** Refers to types of data representation; continuous space involves a range of values, while discrete space consists of distinct, separate values like words in a sentence.
- **Wasserstein GAN (WGAN):** A GAN variant that uses the Wasserstein distance to improve the stability of the training process and mitigate mode collapse.
- **Gradient Penalty:** A regularization technique used in Wasserstein GANs to enforce the Lipschitz constraint, ensuring smoother and more stable training.
- **Binary Neural Network (BNN):** A type of neural network where the weights and activations are constrained to binary values (typically -1 and 1). This approach significantly reduces memory usage and computational complexity, making BNNs highly efficient for deployment on hardware with limited resources, such as mobile devices and embedded systems.

Conflicts of Interest

The authors declare no conflict(s) of interest.

Acknowledgments

I would like to acknowledge the support from the Polygence Program for making this possible.

References

- 1 Rosa, Gustavo and J. Papa, *A Survey on Text Generation Using Generative Adversarial Networks*.
- 2 A. A. ABRO, *Natural Language Processing Challenges and Issues: A Literature Review*, <https://doi.org/10.35378/gujs.1032517>., Accessed 8 Feb. 2024.
- 3 L. James, *NLP Problems: 7 Challenges of Natural Language Processing*, www.metadialog.com/blog/problems-in-nlp/#:~:text=Dealing%20with%20ambiguity%2C%20idiomatic%20phrases., Accessed 8 Feb. 2024.
- 4 R. Wolff, *Natural Language Processing (NLP): 7 Key Techniques*.
- 5 *5, Navigating through Text Generation Challenges and Solutions*, aicontentfy.com/en/blog/navigating-through-text-generation-challenges-and-solutions#:~:One%20of%20the%20main%20challenges., AIContentfy, 1 Sept. 2023,.
- 6 *6, Text Data Augmentation in Natural Language Processing with Textattack*, www.analyticsvidhya.com/blog/2022/02/text-data-augmentation-in-natural-language-processing-with-textattack/.
- 7 V. Dey, *Beginner's Guide to Generative Adversarial Networks (GANs)*, Accessed 8 Feb. 2024.
- 8 I. Goodfellow, *Generative Adversarial Nets*.
- 9 N. Khan, M. Tauseef, R. Ghosh and N. Sarkar, *A Novel Loss Function Utilizing Wasserstein Distance to Reduce Subject-Dependent Noise for Generalizable Models in Affective Computing*, <https://arxiv.org/abs/2308.10869>, arXiv.org, 2023.
- 10 X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang and S. Smolley, *Least Squares Generative Adversarial Networks*, <https://arxiv.org/abs/1611.04076>, arXiv:1611.04076 [cs], Apr. 2017, Available:.
- 11 W. Kim, *Collaborative Training of GANs in Continuous and Discrete Spaces for Text Generation*.
- 12 Y. Wang and S. Zou, *Policy Gradient Method For Robust Reinforcement Learning*, <https://arxiv.org/abs/2205.07344>, arXiv.org, May 15, 2022.
- 13 C.-I. Kim, M. Kim, S. Jung and E. Hwang, *Simplified Fréchet Distance for Generative Adversarial Nets*, <https://doi.org/10.3390/s20061548>.
- 14 A. Celikyilmaz, E. Clark and J. Gao, *Evaluation of Text Generation: A Survey*, <https://arxiv.org/abs/2006.14799>, arXiv:2006.14799 [cs], May 2021, Available:.
- 15 M. Hossam, *Text Generation with Deep Variational GAN*.
- 16 D. Kingma and M. Welling, *An Introduction to Variational Autoencoders*, <https://doi.org/10.1561/22000000056>.
- 17 X. Li, *Feature-Aware Conditional GAN for Category Text Generation*, <https://doi.org/10.1016/j.neucom.2023.126352>., Accessed 28 Oct. 2023.
- 18 A. Santoro, *Relational recurrent neural networks*, <https://arxiv.org/abs/1806.01822>, arXiv.org, 2018.

-
- 19 E. Jang, S. Gu and B. Poole, *Categorical Reparameterization with Gumbel-Softmax*, <https://arxiv.org/abs/1611.01144v5>, arXiv.org, 2016.
 - 20 D. Zhu, H. Yao, B. Jiang and P. Yu, *Negative Log Likelihood Ratio Loss for Deep Neural Network Classification*, <https://arxiv.org/abs/1804.10690>, arXiv.org, Apr. 27, 2018.
 - 21 D. Kuznetsov, *Text Generation with Gan Networks Using Feedback Score*, Accessed 8 Feb. 2024.
 - 22 T. Zhang, V. Kishore, F. Wu, K. Weinberger and Y. Artzi, *BERTScore: Evaluating Text Generation with BERT*, <https://arxiv.org/abs/1904.09675>, arXiv:1904.09675 [cs], Feb. 2020, Available:.
 - 23 M. Youssef Kossale and A. Darouichi, *Mode Collapse in Generative Adversarial Networks: An Overview*, <https://doi.org/10.1109/icoa55659.2022.9934291>.
 - 24 K. Mangalam and R. Garg, *Overcoming Mode Collapse with Adaptive Multi Adversarial Training*, <https://arxiv.org/abs/2112.14406>, arXiv.org, 2021.
 - 25 T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, *Improved Techniques for Training GANs*, <https://arxiv.org/abs/1606.03498v1>, arXiv.org, 2016.
 - 26 C. Papadimitriou, K. Karamanos, F. Diakonou, V. Constantoudis and H. Papageorgiou, *Entropy analysis of natural language written texts*, <https://doi.org/10.1016/j.physa.2010.03.038>.
 - 27 Z. Yu, L. Jaw, Z. Hui and B. Kian, *Fine-tuning Language Models with Generative Adversarial Reward Modelling*, <https://arxiv.org/abs/2305.06176>, arXiv.org, 2023.
 - 28 A. Galassi, M. Lippi and P. Torrioni, *Attention in Natural Language Processing*, <https://doi.org/10.1109/tnnls.2020.3019893>.
 - 29 E. Muszyńska and A. Copestake, *Realization of long sentences using chunking*, <https://doi.org/10.18653/v1/w17-3533>.
 - 30 C. Yuan and S. Agaian, *A comprehensive review of Binary Neural Network*, <https://doi.org/10.1007/s10462-023-10464-w>.