Enhancing GAN for Improved Text Data Augmentation

Matheus Muniz Damasco
Computational Modelling
Federal University of Juiz de Fora
Juiz de Fora, Brasil
matheus.damasco@estudante.ufjf.br

***Abstract*—The increasing demand for high-quality textual data for training large language models has created a critical need for effective data augmentation techniques. This paper presents and evaluates a Generative Adversarial Network (GAN) based on the SeqGAN architecture for generating synthetic text. The model, implemented in PyTorch and trained on the COCO Captions dataset, employs a two-phase training strategy: an initial pre-training of the generator via Maximum Likelihood Estimation (MLE), followed by adversarial fine-tuning using a policy gradient from Reinforcement Learning. Furthermore, we address the critical challenge of generation diversity by showing that the model produces novel sentences, with a plagiarism rate of less than 1% against the training corpus and over 98% internal originality. These findings validate the proposed approach as a powerful method for generating high-quality, diverse synthetic data, offering a promising solution to mitigate data scarcity in Natural Language Processing.**

***Index Terms*—Generative Adversarial Networks, Data Augmentation, Text Generation, Natural Language Processing, Deep Learning**

I. Introduction

The rapid progress in technology over the last few years, encompassing both hardware and software, has propelled the expansion of Artificial Intelligence, which in turn has fostered the development of Natural Language Processing. Natural Language Processing (NLP) is an area of Artificial Intelligence that enables computers to understand, interpret, and generate human language texts. Based on Machine Learning and Deep Learning techniques, NLP analyzes large volumes of textual data to identify patterns, extract information, and perform tasks such as translation, classification, and sentiment analysis [1]. Originally based on rules and heuristics, it evolved into statistical and neural methods capable of efficiently handling unstructured data [2].

Recent advancements in the field of Natural Language Processing are headlined by Large Language Models (LLMs). The success of Large Language Models (LLMs) is inherently linked to the availability of vast, diverse, and high-quality data for training. However, the growth rate of high-quality data is significantly outpaced by the expansion of training datasets, leading to a looming data exhaustion crisis. This underscores the urgent need to enhance data efficiency and explore new data sources [3]. In this context, data generation—primarily through data augmentation and synthesis—has emerged as a promising solution [3]. A variety of techniques support this goal, from simpler methods like EDA [4] and Back Translation [5] to more complex deep learning approaches based on Variational Autoencoders (VAEs) [6], Generative Adversarial Networks (GANs) [7], and even LLMs themselves [8] and Diffusion Models [9]. Therefore, this paper proposes and evaluates a novel Generative Adversarial Network (GAN) designed specifically for textual data augmentation.

The remainder of this paper is organized as follows. Section 2 further describes the problem. Section 3 reviews related work in textual data augmentation, with a focus on GANs. Section 4 describes the architecture and training process of our proposed GAN. Section 5 presents the experimental setup, results, and a comparative analysis against baseline methods. Finally, Section 6 concludes the paper and discusses potential avenues for future work.

II. Description of the Problem

As introduced in the previous section, the scalability of modern Large Language Models (LLMs) is fundamentally threatened by a scarcity of training data. This is not a distant, theoretical issue, but an imminent bottleneck. In the article ”Will we run out of data? Limits to the scalability of LLM based on human-generated data”, it was estimated that the stock of high-quality human-generated public texts will reach its midpoint of depletion in 2028, with a high probability of complete depletion by 2032 [10]. This looming ”data exhaustion” makes the development and refinement of data augmentation techniques not just beneficial, but critical for the continued progress of the field.

The immense pressure on data resources is a direct consequence of the empirical ”Scaling Laws” that have governed the development of LLMs. These principles demonstrate a predictable, power-law relationship between a model's performance and increases in model size, compute, and dataset size. This has fueled a paradigm where creating larger models necessitates acquiring ever-larger volumes of text, thus driving the insatiable demand that is rapidly consuming our finite stock of human-generated data [11].

The practical consequences of this data scarcity are profound. It exacerbates the existing digital divide, as the lack of large-scale datasets for low-resource languages prevents the benefits of advanced NLP from being universally accessible, a central challenge in the era of large pre-trained models [12],

[13]. Furthermore, it creates significant barriers in specialized domains like medicine, law, and finance, where data is inherently scarce due to privacy regulations (such as HIPAA), proprietary restrictions, and the high cost of expert annotation [14].

This leads to a seemingly obvious solution: using data generated by previous models to train future ones. However, recent research has revealed a critical flaw in this recursive approach, a phenomenon termed ”Model Collapse”. Studies have shown that training models on synthetic data causes them to progressively forget rare events, lose diversity, and amplify their own biases over successive generations, leading to a degenerative feedback loop. This ”Curse of Recursion” highlights that the challenge is not merely to generate more data, but to generate synthetic data that retains the quality and diversity of human-created text [15].

Therefore, the problem to be solved is twofold: there is a quantitative crisis defined by the finite supply of new, high-quality human data, and a qualitative crisis defined by the risks of model degradation when using naive data generation methods. This context highlights the need to develop sophisticated data augmentation techniques.

III. Related Work

Introduced in 2014, the method known as Generative Adversarial Networks (GANs) consists of two neural networks competing against each other: a Generator and a Discriminator. The Generator network aims to create synthetic data that mimics the real training data, while the Discriminator seeks to distinguish between real and generated samples. During training, both networks are continuously refined: the Generator improves its ability to create realistic data, and the Discriminator becomes more effective at identifying the generated samples. This competitive process results in the generation of increasingly convincing synthetic data [7]. The practical impact of Generative Adversarial Networks is vast, with applications spanning numerous fields, but the most prominent successes have been achieved in Computer Vision, where GANs are employed for tasks far beyond simple image generation. These include image-to-image translation, image inpainting, and video processing. Beyond visual data, their utility extends to Natural Language Processing for text generation, as well as to cross-domain applications like data augmentation to improve the robustness of other models. This innovative architecture’s ability to generate realistic content has attracted significant and sustained research interest [16].

Adapting the original GAN framework to the text domain presents significant challenges not found in image generation. The primary issue stems from the discrete nature of text: while images are sampled from continuous pixel distributions, text requires sampling discrete tokens from a finite vocabulary. Furthermore, generating coherent text requires capturing complex properties like long-range dependencies and hierarchical structure, issues that are less prevalent in pixel-level image synthesis [17]. Consequently, these combined challenges necessitated the development of entirely new architectural approaches, beginning in 2017, designed to solve these textual data generation problems. The first GANs developed with a focus on textual data are SeqGAN, which framed the generator as a reinforcement learning agent [18]; MaliGAN, which proposed a modified objective function [19]; and RankGAN, which replaced the binary discriminator with an adversarial ranker [20]. While these foundational models demonstrated that text generation with GANs was possible, their success inspired a new line of research focused on the task of data augmentation.

IV. Methodology

The methodology employed in this work is an implementation of the SeqGAN (Sequence Generative Adversarial Network) architecture [18], developed using the PyTorch framework. It consists of two main components trained adversarially: a Generator (G) and a Discriminator (D). The entire training process is divided into two distinct phases: a pre-training phase with Maximum Likelihood Estimation (MLE) [21] and a final adversarial training phase using a Policy Gradient [22] from Reinforcement Learning [23].

A. Generator Architecture

The Generator was designed as a recurrent neural network (RNN) [24] based on Long Short-Term Memory (LSTM) [25] units, tasked with generating text sequences token by token. Its architecture is composed of:

- An Embedding layer that maps input tokens into dense vectors.
- A single-layer LSTM that processes the embedded tokens sequentially.
- A final Linear layer with a Log-Softmax activation function, which outputs a probability distribution over the entire vocabulary for the next token in the sequence.

B. Discriminator Architecture

The Discriminator was implemented as a text classifier designed to differentiate between real sequences from the dataset and synthetic sequences from the Generator. Its architecture is composed of:

- An Embedding layer with a vector dimension.
- A 3-layer bidirectional LSTM with a hidden state dimension. The bidirectional nature allows the model to capture context from both forward and backward directions. A dropout was applied to prevent overfitting.
- A final Linear layer followed by a Sigmoid activation function, which outputs a single scalar probability score between 0 (fake) and 1 (real).

C. Training Procedure

The training process was strategically divided into two main phases to ensure stability and effectiveness:

1) Maximum Likelihood Estimation (MLE) Pre-training: To provide a robust initialization for the Generator, it was first pre-trained using a standard language modeling objective. In this phase, the Generator is trained solely to minimize the Negative Log-Likelihood (NLL) loss, which teaches it to predict the next token in a real sentence. This step is crucial for the Generator to learn the basic grammar and structure of the language before facing the Discriminator.

2) Adversarial Training: After pre-training, the models enter a competitive training phase, following the SeqGAN algorithm. This phase iterates between two steps:

- Training the Discriminator: The Discriminator is trained on a mixed batch of real sentences (labeled as 1) and synthetic sentences from the Generator (labeled as 0). Its objective is to minimize the Binary Cross-Entropy (BCE) loss [26], thus improving its ability to distinguish real from fake.
- Training the Generator: The Generator is updated using a policy gradient method from Reinforcement Learning. It generates a batch of sequences, and the Discriminator provides a reward score for each sequence. This reward is then used to update the Generator's weights, encouraging it to produce sequences that are more likely to be classified as real by the Discriminator

D. Dataset

The experiments were conducted using the Microsoft COCO (Common Objects in Context) Image Captions dataset [27], a widely recognized benchmark for image captioning and text generation tasks. This dataset consists of everyday scenes, where each image is paired with five distinct descriptive captions provided by human annotators. The captions are typically factual, declarative sentences describing the objects and actions within the image, providing a rich and well-defined corpus for training generative models. The official dataset is made publicly available through the official project website [1].

E. Parameters

The hyperparameters used throughout the experiments are detailed below:

- **Parameters:**
  - Generator Embedding Dimension: 150
  - Generator Hidden Dimension: 150
  - Discriminator Embedding Dimension: 150
  - Discriminator Hidden Dimension: 150
  - Discriminator Dropout Rate: 0.2
- **Training Parameters:**
  - MLE Pre-training Epochs: 30
  - Adversarial Training Epochs: 25
  - Generator Optimizer: Adam
  - Discriminator Optimizer: Adagrad
- **Dataset and Batching Parameters:**
  - Vocabulary Size: 4980
  - Maximum Sequence Length: 20
  - Batch Size: 1000

V. RESULTS

The evaluation of the proposed model was conducted in distinct phases, mirroring the training procedure to effectively gauge the contribution of each component. The model's performance was quantitatively assessed using Negative Log-Likelihood (NLL) [21] and Bilingual Evaluation Understudy (BLEU) scores [28].

Initially, the Generator was pre-trained for 30 epochs using Maximum Likelihood Estimation (MLE). This phase proved crucial, reducing the average test NLL from an initial 8.5451 to 1.9609. This significant decrease indicates that the Generator successfully learned the fundamental grammar, structure, and vocabulary distribution of the COCO Captions dataset [27] before the adversarial stage. Concurrently, the Discriminator was pre-trained and achieved a final validation accuracy of 80.4%, demonstrating its capability to generated text sequences.

Following the pre-training, the model underwent 25 epochs of adversarial training. To evaluate the quality of the generated sentences, we computed BLEU scores (n-gram precision for n=2 to 5) against the test set. We compared the final adversarially trained model against the MLE-trained Generator and a baseline calculated by comparing samples from the training set against the test set. Furthermore, we investigated the effect of temperature scaling on generation quality by adjusting the sampling degree (where a degree of 1.5 corresponds to a lower temperature, favoring more likely tokens).

The results are summarized in Table I. The MLE-trained generator with standard sampling (degree=1.0) produced text with lower quality than the training data baseline. However, by adjusting the sampling degree to 1.5, its performance improve, surpassing the baseline across all BLEU scores. This highlights the sensitivity of generation quality to the sampling strategy.

The adversarially trained GAN model with standard sampling (degree=1.0) showed a slight improvement over its MLE counterpart in BLEU-4 and BLEU-5 scores, suggesting that adversarial fine-tuning enhances sequence coherence. The most significant finding emerged from the combination of adversarial training and a sampling degree of 1.5.

TABLE I
BLEU SCORE EVALUATION OF GENERATED TEXT

Model Configuration	**BLEU-2**	**BLEU-3**	**BLEU-4**	**BLEU-5**
Training Data (Baseline)	0.5499	0.4369	0.3173	0.2108
MLE (degree=1.0)	0.5227	0.3826	0.2474	0.1487
MLE (degree=1.5)	0.5601	0.4719	0.3652	0.2528
GAN (degree=1.0)	0.5124	0.3824	0.2608	0.1641
GAN (degree=1.5)	**0.5646**	**0.4987**	**0.4120**	**0.3072**

Finally, to address the qualitative aspects of data generation and the risk of "Model Collapse" [15], we assessed the novelty of the generated samples. The final model exhibited a plagiarism rate of only 0.67% when compared against the 80,000 sentences in the training corpus. Furthermore, an

[1] COCO Image Captions dataset: https://cocodataset.org/

internal originality check revealed that 98.4% of the generated samples were unique, confirming that the model produces diverse and novel text rather than merely memorizing training examples. For instance, a representative sample generated by the final model is: *"A man takes a picture of a slice on top of a table."* This sentence is grammatically correct and coherent.

VI. Conclusions

This paper presented and evaluated a GAN architecture for the task of textual data augmentation. By training the model on the Microsoft COCO Captions dataset, we demonstrated that generative adversarial networks, when properly configured, are a powerful tool for creating high-quality, synthetic text.

Our primary finding is that the combination of adversarial training with a policy gradient method and an adjusted sampling temperature (degree=1.5) significantly enhances the quality of generated data. The final model outperformed a strong MLE baseline and even the statistical similarity between samples of the original dataset, achieving a BLEU-4 score of 0.4120. This result suggests that the adversarial process successfully guides the generator to produce more coherent and human-like sentence structures that go beyond simple probabilistic token prediction. Furthermore, our analysis confirmed that the generated data is novel, with a plagiarism rate below 1% against the training set and high originality, thereby addressing the critical challenge of generating diverse data to mitigate model collapse.

Despite these promising results, this study has limitations. The evaluation was confined to a single, relatively structured dataset of image captions. The model's performance on more complex, long-form, or domain-specific text remains to be explored. Additionally, while BLEU is a standard metric, it primarily measures n-gram overlap and may not fully capture semantic correctness or factual accuracy.

Future work will proceed in two main directions. First, we intend to evaluate our GAN using a broader range of datasets and employing metrics beyond BLEU. Second, we will assess the GAN's performance on languages such as Portuguese, where data is scarcer and the language exhibits greater complexity than English.

References

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing (3rd ed. draft)*. Prentice Hall, 2020. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

[2] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[3] K. Wang, Z. Chen, Z. Wang, F. Jiao, C. Yi, H. Wang, Z. Xu, R. Fu, X. Jiang, J.-J. Huang, X. Zhang, and Y. You, "A survey on data synthesis and augmentation for large language models," 2024. [Online]. Available: https://arxiv.org/abs/2410.12896

[4] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6382–6388. [Online]. Available: https://aclanthology.org/D19-1670/

[5] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.–Nov. 2018, pp. 489–500. [Online]. Available: https://aclanthology.org/D18-1045/

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022. [Online]. Available: https://arxiv.org/abs/1312.6114

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[10] P. Villalobos, A. Ho, J. Sevilla, T. Besiroglu, L. Heim, and M. Hobbhahn, "Will we run out of data? limits of llm scaling based on human-generated data," *arXiv preprint arXiv:2211.04325v2*, 2022, revised June 4, 2024. [Online]. Available: https://arxiv.org/abs/2211.04325v2

[11] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," 2020.

[12] P. Joshi, S. Santy, A. Budhiraja, K. Krishna, and M. Bansal, "The state and fate of nlp in the era of pre-trained language models," 2020.

[13] L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P.-S. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh, Z. Kenton, S. Subramanian, d. W. H. Owen an, T. Pohlen, Z. Fu, C. Biles, and I. Gabriel, "A taxonomy of risks posed by language models," 2021.

[14] S. Wu, K. Roberts, S. Datta, J. Du, Z. Ji, Y. Si, S. Soni, Q. Wang, Q. Wei, Y. Xiang *et al.*, "A survey on recent advances in nlp for electronic health records," *npj Digital Medicine*, vol. 3, no. 1, pp. 1–12, 2020.

[15] I. Shumailov, Z. Shumaylov, Y. Zhao, Y. Gal, N. Papernot, and R. Anderson, "The curse of recursion: Training on generated data makes models forget," 2023.

[16] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3313–3332, 2021.

[17] M. Caccia, L. Caccia, W. Fedus, H. Larochelle, J. Pineau, and L. Charlin, "Generative adversarial networks for text generation: A survey," 2020.

[18] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017, pp. 2852–2858.

[19] T. Che, Y. Li, R. Zhang, R. D. Hjelm, W. Li, Y. Bengio, and L. Nie, "Maximum-likelihood augmented discrete generative adversarial networks," 2017.

[20] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun, "Adversarial ranking for language generation," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[21] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 222, no. 594-604, pp. 309–368, 1922.

[22] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[23] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[24] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.