AI-generated text vs Human-written text detection using a Generative Adversarial Network (GAN)

A Thesis
Presented to the
Department of Computer Science
College of Information and Computing Sciences
University of Santo Tomas

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science in Computer Science

By

Mallari, Mico Angelo
Santos, Aaliyah Makayla
Tolentino, Rafael Gerard
Vargas, Justin Andrie

Adviser:

Asst. Prof. Rochelle Lynn Lopez, DT

November 2024

Approval Sheet

Thesis Title: AI-generated text vs Human-written text detection using a Generative Adversarial Network (GAN)

Researchers:

1. **Mico Angelo C. Mallari**
2. **Aaliyah Makayla Santos**
3. **Rafael Gerard A. Tolentino**
4. **Justin Andrie S. Vargas**

In partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Science,** the thesis mentioned above, has been adequately prepared and submitted by above-mentioned researchers. This thesis was duly defended in an oral examination before a duly constituted tribunal on 09/30/2024 with a grade of

____
Asst. Prof. Cecil Jose Delfinado
Panel Member

____
Asst. Prof. Darlene Alberto
Panel Member

____
Asst. Prof., Charmaine S. Ponay
Thesis Coordinator

____
Asst Prof. Cherry Rose R. Estabillo
Panel Member

____
Asst. Prof. Rochelle Lynn Lopez, DT
Thesis Adviser

Accepted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

____
Asst Prof. Cherry Rose R. Estabillo
Program Chair
Department of Computer Science
College of Information and Computing Sciences

University of Santo Tomas
College of Information and Computing Sciences
Department of Computer Science

Certificate of Authenticity and Originality

We, the authors of this thesis, **“AI-generated text vs Human-written text detection using a Generative Adversarial Network (GAN)”**, hereby certify and vouch that the contents of this research work is solely our own original work; that no part of this work has been copied nor taken without due permission or proper acknowledgment and citation of the respective authors; that we are upholding academic professionalism by integrating intellectual property rights laws in research and projects as requirements of our program.

If found and proven that there is an attempt or committed infringement of copyright ownership, we are liable for any legal course of action sanctioned by the University and the Philippine laws.

Mallari, Mico Angelo

Santos A.

Santos, Aaliyah Makayla

Tolentino, Rafael Gerard

Vargas, Justin Andrie

Program: BS Computer Science

Date: November 30, 2024

Table of Contents

List of Figures

List of Tables

Chapter I : The Problem and Its Background

A. Introduction

In an era where Artificial Intelligence (AI) and Large Language Models (LLMs) have been developing at an alarming rate, we are now at a point where computers can now generate convincing human-like text. The advent of generative AI has brought about technological breakthroughs in computer science. Yet, it also brings ethical and societal issues, such as misleading AI-generated news and online content, which may sometimes endorse conflict between individuals online and offline. The challenge of distinguishing artificial intelligence-generated writing from human-made content has become more challenging due to the rapid developments in machine learning and computational languages.

Since machine-generated text detection has become an area of interest, many challenges have been presented in testing and comparing different deep learning algorithms. Although current machine learning models can generate text, only a few techniques can consistently and accurately differentiate between text generated by these models and text written by humans. While several existing solutions may be helpful, they cannot handle various complex generative models and diverse language styles. To address these problems, different researches have been explored and analyzed relating to different text detection models and deep learning techniques.

This paper focused on the use of deep learning techniques to detect AI-generated text using RoBERTa and Generative Adversarial Network (GAN).

Generative Adversarial Networks (GANs) are a powerful tool for text data augmentation, as noted by Silva et al. (2023). GANs have been applied to tasks like sentiment analysis, low-resource language generation, and hate speech detection, addressing data scarcity with realistic synthetic samples. Despite challenges in generating coherent text due to the structured nature of language, GAN-based augmentation significantly enhances NLP model robustness and generalization, particularly for small or imbalanced datasets.

In a study by Hu et al. (2023), they used a Generative Adversarial Network (GAN) model to produce synthetic text based on the everyday datasets they have collected, such as Xsum, TOEFEL, Reddit WritingPrompts, SQuAD, and WP. They detected AI-generated and Human-written texts using a paraphraser and detector based on T5-large and RoBERTa-large, respectively, using their RADAR framework. The results indicate that without a paraphraser, OpenAI's RoBERTa model achieved 90% AUROC score over RADAR's 85.6% AUROC score. However, when implemented with seen and unseen paraphrasers, RADAR's model achieved 92.2% and 85.7% AUROC scores, respectively, over OpenAI's RoBERTa model, which yielded only 80.6% and 62.7% AUROC scores.

To the best of the researchers' capabilities, there are minimal studies that integrate Generative Adversarial Networks (GAN) with Pre-trained Language Models (PLM) in the context of AI-generated text detection and data augmentation. Therefore, the researchers of this study opted to contribute in this area.

B. Background of the Study

Existing studies used deep learning, specifically transformer-based models, for the classification and detection of AI-generated text from human-written text. In the experiment by Capobianco et al. (2024), they compared the performance of BERT and RoBERTa with frozen and unfrozen parameter variations using the HC3 dataset, which comprises human and AI question responses. Using the RoBERTa language model and unfrozen weights trained on 5% of the dataset, a score of 0.98 was yielded for testing accuracy, 0.97 for precision, 0.98 for recall, and 0.98 for F1-score. The results showed that RoBERTa could yield a higher score than using the BERT model. Thus, it can be concluded that RoBERTa is a more effective natural language processing model for classifying the HC3 dataset than BERT.

In another comparative study by Wang et al. (2023), it was noted that the BERT model had an accuracy rate of more than 87%, while the improved RoBERTa model had an accuracy rate of 98% in classifying AI-generated news.

Another example is the study of Croce et al. (2020). In this research, SS-GAN or Semi-Supervised Generative Adversarial Networks extend the implementation of BERT for AI-generated text detection, reaching 62.67% and 60.45% F1 scores. In comparison, BERT alone reaches 48.35% and 42.41% F1 scores for mismatched and matched data, respectively, showing how integrating Generative Adversarial Network (GAN) improves overall performance in AI-generated text detection.

In a study by Surbhi Sharma (2024), the GAN-RoBERTa framework achieved an F1 score of 83.26% and an accuracy of 73.6% when trained on a mix of labeled and unlabeled datasets for AI-generated text detection. This study highlighted the use of GANs to generate synthetic text samples, which enhanced the RoBERTa model's robustness in distinguishing human-written text from machine-generated content.

Guo et al. (2023) presented the Human ChatGPT Comparison Corpus or HC3 dataset specifically for classifying AI-generated and human-written responses. It comprises several subcategories of questions and reactions, including financial, medical, open question-answer, Reddit prompts, and wiki statements. Similarly, several studies use multiple datasets that cover a wide range of domains for robust performance evaluation. In the study by Mitchell et al. (2023), they evaluated and compared the performance of different models such as DetectGPT, RoBERTa-base, and RoBERTa-large when using the Xsum, SQuAD, and Reddit WritingPrompts dataset separately. Their findings showed that RoBERTA-base and RoBERTa-large yielded higher average AUROC scores for the SQuAD (Wikipedia text) and Reddit WritingPrompts (creative text) datasets. In contrast, DetectGPT yielded a higher average AUROC score for the Xsum (news text) dataset.

In another study by Chen et al. (2023), the authors fine-tuned RoBERTa, GLTR, and STADEE models and used HC3-Chinese, ChatGPT-CNews, and CPM-CNews datasets. For RoBERTa, they used a batch size of 48, a learning rate of 5×e−5, an AdamW optimizer, and fine-tuned in 2 epochs. This results in 97.96 and 98.37 F1 scores for validation and testing sets, respectively, which puts RoBERTa on top of the other models (GLTR yielded 78.19 and 77.77 while STADEE yielded 87.65 and 87.05) for comparison.

Similar to the study by Gaggar et al. (2023), they experimented with different transformer-based models for AI-generated text detection, such as SVM, RoBERTa-base, and RoBERTa-large. The results indicate that RoBERTa-base exhibits elevated performance compared to SVM, while RoBERTa-large further enhances this performance across the train, test, and validation sets. However, the author pointed out that when selecting datasets, future work can aim to add a broader range of text from more sources for a variety of text structures and styles. They stated that this approach will make the data more versatile, suitable for different fields, and capable of handling texts of different lengths. With that in mind, the researchers are encouraged to utilize a collection of datasets among various domains, such as creative writing, Wikipedia texts, and financial, medical, academic essays, and many more. The authors also recommended integrating and testing more advanced models to enhance classification performance.

Given that multiple studies have tackled AI-generated text detection with numerous and mixed datasets for performance evaluation, this study proposed to introduce Generative Adversarial Network (GAN) to the existing model used in the study of Gaggar et al. (2023) where RoBERTa was also used. Since Generative Adversarial Network (GAN) and generating synthetic AI-generated text has seen minimal application in AI-generated text detection, the proposed method in this study for improving the AUC-ROC score is to integrate a Generative Adversarial Network model on the existing model used in the study of Gaggar et al. (2023) wherein their RoBERTa was fine-tuned for AI-generated text detection. Using GAN along with a pre-trained RoBERTa model and a dataset consisting of many sources and domains will comparatively show how Adversarial Training and Data Augmentation impacts performance scores on AI-generated text detection.

C. Theoretical Framework

The following section presents the existing models and frameworks that served as the foundation of the study.

a. RoBERTa

RoBERTa is a pre-trained case-sensitive model on the English language. This model was presented by researchers at Facebook and Washington University, in which the goal of this paper was to optimize the training of BERT architecture to take less time during pre-training. Its pre-training involved 160GB of self-supervised data (Liu et al. 2019), 10 times larger than the dataset used to train BERT. Additionally, RoBERTa used masked language modeling (MLM), which takes a sentence and masks 15% of the words in the input at random. The model then runs the complete masked sentence through the model and needs to predict the masked words. This is unlike autoregressive models like GPT, which internally mask the future tokens, or conventional recurrent neural networks (RNNs), which typically see the words one after the other. It enables the model to pick up a sentence's bidirectional representation.

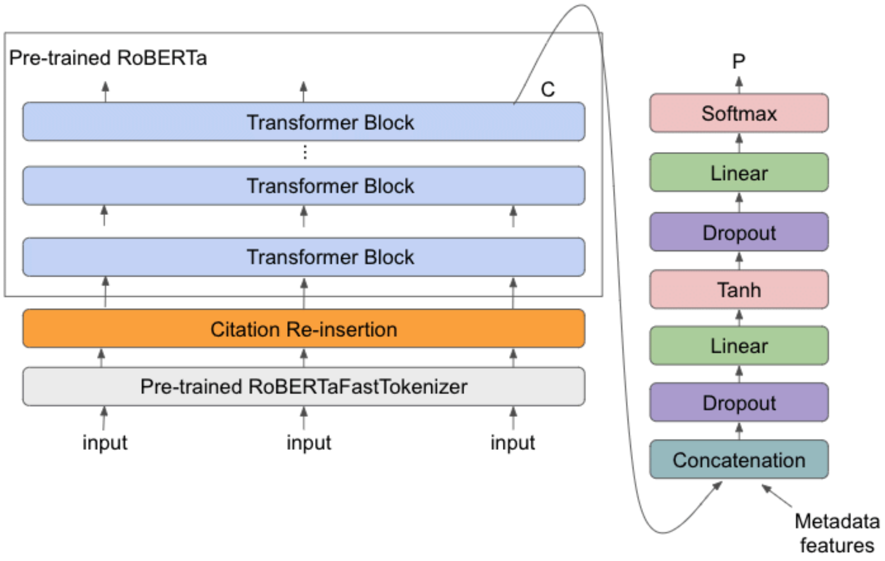


Figure 1.1 RoBERTa Architecture

According to an analysis by Sharma (2022), the RoBERTa is a reimplementation of BERT with some modifications to the key hyperparameters and minor embedding tweaks, therefore, it shares the same architecture as BERT. It uses a byte-level BPE as a tokenizer (similar to GPT-2) and a different pre-training scheme. Also, RoBERTa is trained for longer sequences, the number of iterations is increased from 100K to 300K and then further to 500K. RoBERTa uses larger byte-level BPE vocabulary with 50K subword units instead of character-level BPE vocabulary of size 30K used in BERT. Larger mini-batches and learning rates are used in RoBERTa's training as well.

It has been demonstrated that RoBERTa performs better than BERT and other cutting-edge models on a range of natural language processing tasks, such as text classification, question answering, and language translation. It is now a preferred option for both academic study and commercial applications. It has also served as the foundation model for numerous other effective NLP models. Given multiple studies showing how successful RoBERTa is with NLP tasks such as AI-generated text detection, it is proven that this pre-trained language model is superior in performance compared to many others.

b. GAN

Generative Adversarial Networks (GANs) were developed back in 2014 by Ian Goodfellow and his teammates. GAN is basically an approach to generative modeling that generates a new set of data based on training data that replicates the content of the training data. GANs have two main blocks (generator and discriminator) which compete with each other and are able to capture, copy, and analyze the variations in a dataset.

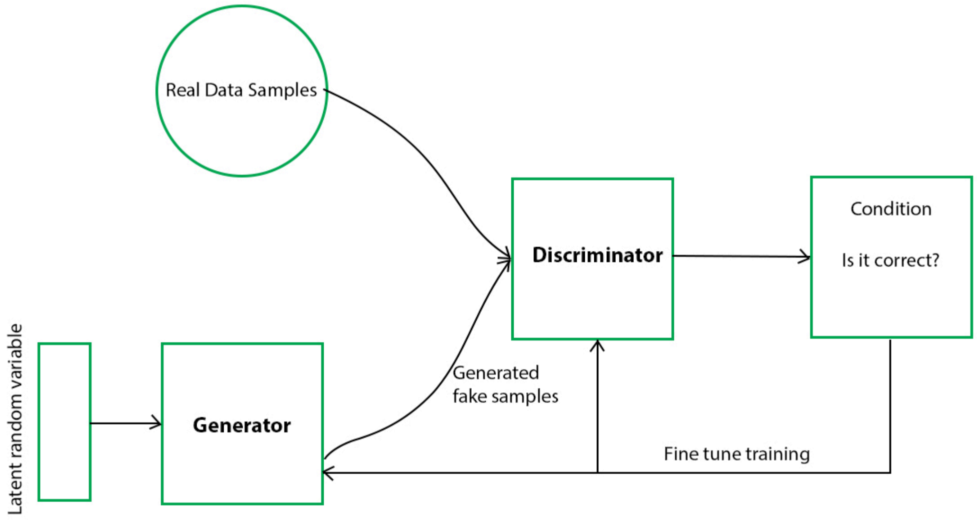


Figure 1.2 GAN Architecture

In a research by Agrawal (2024), he stated that the generative model captures the distribution of data and is trained in such a manner to generate the new sample that tries to maximize the probability of the discriminator to make a mistake (maximize discriminator loss). The discriminator on other hand is based on a model that estimates the probability that the sample it receives is from training data not from the generator and tries to classify it accurately and minimize the GAN accuracy. Hence the GAN network is formulated as a minimax game where the Discriminator is trying to minimize its reward V(D, G) and the generator is trying to maximize the Discriminator loss.

In the study conducted by Croce et al. (2020), they extended BERT by using SS-GANs for the fine-tuning stage. Two elements were added to an already-trained BERT model to modify its fine-tuning: task-specific layers, which are part of the standard BERT fine-tuning, and SS-GAN layers, which allow for semi-supervised learning.

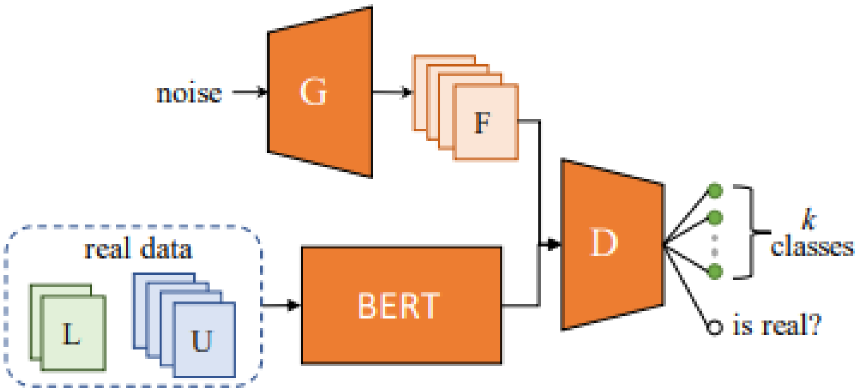


Figure 1.3 GAN-BERT for text classification Architecture (Croce, Castellucci, & Basili, 2020)

Figure 1.2 illustrates how they extend the SS-GAN architecture beyond BERT by including a generator G that acts adversarially and a discriminator D, for example, classification. G creates a vector representation of synthetic text data by implementing it as a Multi-Layer Perceptron (MLP) and accepting as input a 100-dimensional noise vector sampled from a normal distribution. D, on the other hand, is also implemented as an MLP, and it gets input vectors that represent real text examples or the generated synthetic text. The discriminator uses a softmax-activated layer for classification to discriminate between authentic and fraudulent text representations.

In the training process, the discriminator optimizes two competing losses, LD and LG, by classifying real cases into k categories and synthetic instances into a k + 1 category. While unlabeled cases only contribute to LDunsup (unsupervised loss) if incorrectly categorized into the k + 1 category, labeled examples contribute to supervised loss (LDsup). Other than that, contributions from unlabeled samples are hidden. The discriminator is penalized by generated examples if they cannot identify synthetic examples, and vice versa. During discriminator updates, labeled and unlabeled data adjust BERT weights. G is eliminated following training, freeing up BERT for inference. A Generative Adversarial Network (GAN) is a powerful adversarial technique that leverages the generalization of data by generating synthetic data to learn its characteristics accurately.

c. AI-generated Text Detection

A study was conducted by Hu et al. (2023) to improve AI-text detection, they propose RADAR, a framework for training a robust AI-text detector using adversarial learning.

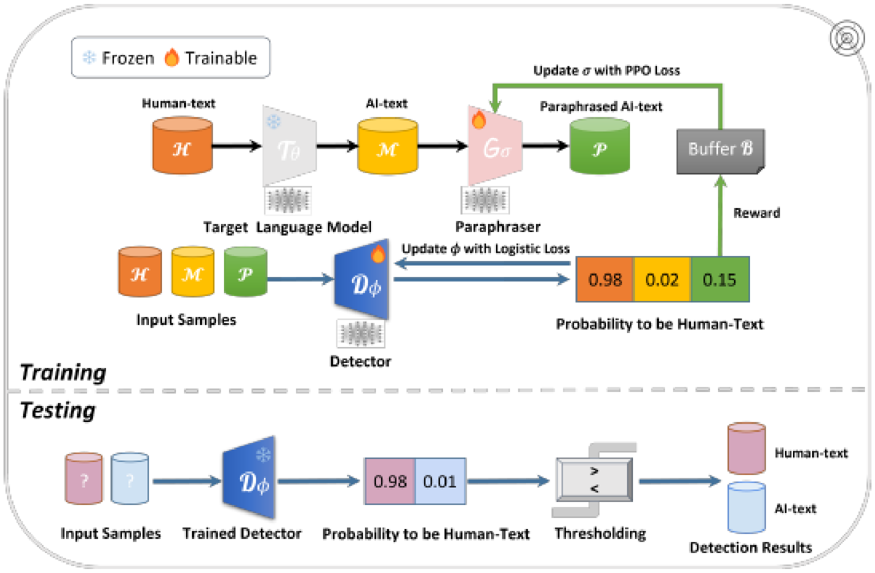


Figure 1.4 RADAR Framework by Hu et al. (2023)

An overview of RADAR is illustrated in Figure 1.4. Their approach is inspired by adversarial machine learning methods like generative adversarial networks (GANs), which train a high-quality generator by inserting a discriminator to create a two-player game. A paraphraser and a detector are introduced as two players in RADAR who have opposing goals. While the detector seeks to improve AI-text detectability, the paraphraser aims to provide realistic content that can avoid detection. Within their framework, different LLMs parametrize the detector and the paraphraser.

To reduce the possibility that the detector will predict an AI text during training, the paraphraser rewrites text from a training corpus (created by a target LLM from a human-text corpus). Meanwhile, the detector attempts to improve detection performance by learning to compare human text vs. AI-text derived from the output of the paraphraser and the training data. Iteratively updating their model parameters, these two players continue until one of their validation losses stabilizes. Their experimental results show that RADAR significantly outperforms existing AI-text detection methods and models, such as OpenAI's RoBERTa model, especially when paraphrasing is in place.

D. Conceptual Framework

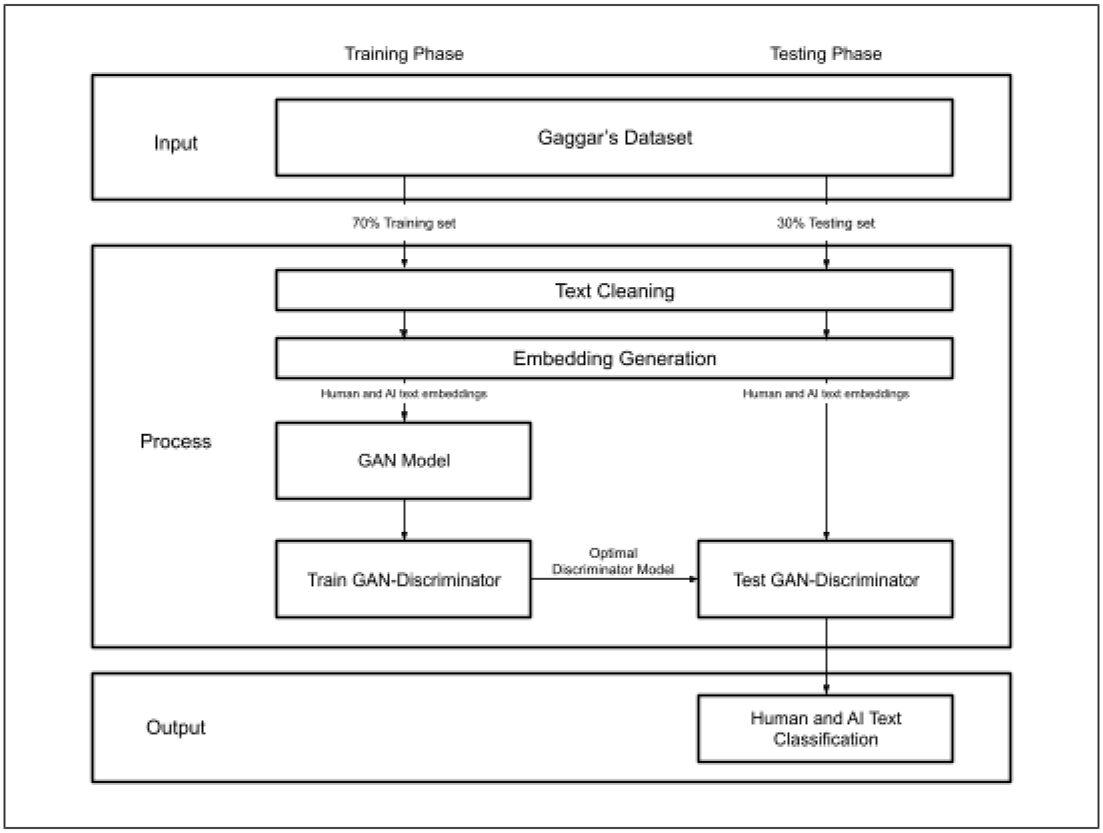


Figure 1.5 Conceptual Framework

This study proposed a three-phase evaluation framework to assess and compare a GAN-based discriminator's ability to classify AI-generated and human-written text against the RoBERTa model used in the study of Gaggar et al. (2023). In the training phase, the dataset is split into 70% training and 30% testing sets. The training set undergoes pre-processing, including text cleaning and embedding generation, which converts the text into numerical representations suitable for input into the GAN model. The GAN model is then trained, focusing on optimizing the discriminator to differentiate between human-written and AI-generated text. This training process produces an optimal discriminator model, fine-tuned to the patterns and features present in the dataset. In the testing phase, the pre-processed testing set is used to evaluate the performance of the trained GAN discriminator. The embeddings from the testing data are fed into the discriminator, which classifies the text samples as either human-written or AI-generated. Evaluation metrics such as accuracy and AUC-ROC are computed to assess the discriminator's classification performance. The final phase involves comparing the text detection performance of the GAN-based discriminator with the RoBERTa model used in Gaggar et al.'s study. By directly comparing the classification metrics of both models on the same dataset, this framework provides insights into the relative effectiveness of GAN-based text classification versus the fine-tuned RoBERTa model. This comparison highlights the potential advantages of using GANs for text detection tasks.

E. Statement of the Problem

The study of Gaggar et al. (2023) can still be improved since their fine-tuned RoBERTa-base model only achieved 95.24% AUC-ROC score on AI-generated text and Human-written text classification on their dataset's test set. Furthermore, their RoBERTa model struggles with nuanced patterns specific to GPT-3.5 Turbo, as the AI sentences in their dataset were rephrased with fixed constraints, limiting diversity. A GAN-based approach addresses this by dynamically generating diverse synthetic text samples, enabling the discriminator to better capture the subtleties of GPT-3.5 Turbo-generated text and improving classification performance. The questions that arise from this study are the following:

1. How can the GAN model be configured to improve text classification performance for AI-generated text detection compared to the model of Gaggar et al.?
2. Will the use of GAN be able to achieve a higher AUC-ROC than what was achieved by Gaggar et al.?
3. How will the proposed GAN model perform on categorized AI-generated and human-written texts across different contexts?

F. Objectives

The study's main objective is to improve the AUC-ROC score of the existing model of Gaggar et al. (2023) by using a Generative Adversarial Network (GAN) using their dataset. Specifically, this study aims:

1. To fine-tune the GAN model to enhance text classification performance and improve upon the model of Gaggar et al.
2. To achieve a higher AUC-ROC in classifying AI-generated text and human-written text compared to the 95.24% that was achieved by Gaggar et al.
3. To evaluate the comparative performance of the GAN model per AI-generated text and human-written text domain.

G. Scope and Limitations

The dataset used in this study was sourced from Gaggar's work, consisting of two categories: AI-generated and human-written text. It contains approximately 400k AI-labeled text and 380k human-labeled text. Since this dataset is already balanced, no additional data balancing was required to ensure fair performance evaluation when detecting human and AI-generated text. Additionally, the dataset did not provide predefined test sets, so the researchers applied their own data splitting strategy to create training, validation, and test sets.

The distribution of texts and domains of the human and AI samples from the dataset are listed in the table below:

Table 1.1 Gaggar's dataset contents

Dataset	Description	Domains
Twitter Sentiment	Short-form text, likely reflecting opinions or sentiments typical of Twitter posts.	Social Media
Football Commentary	Domain-specific commentary text, providing context from sports reporting.	Sports
Project Gutenberg	Literature texts, offering longer-form sentences with a focus on varied literary styles.	Literature
PubMedQA	Medical domain text, including scientific and technical language.	Medical/Scientific
SQuAD (Stanford Question Answering Dataset)	Comprehension and academic-focused text.	Education/Academic

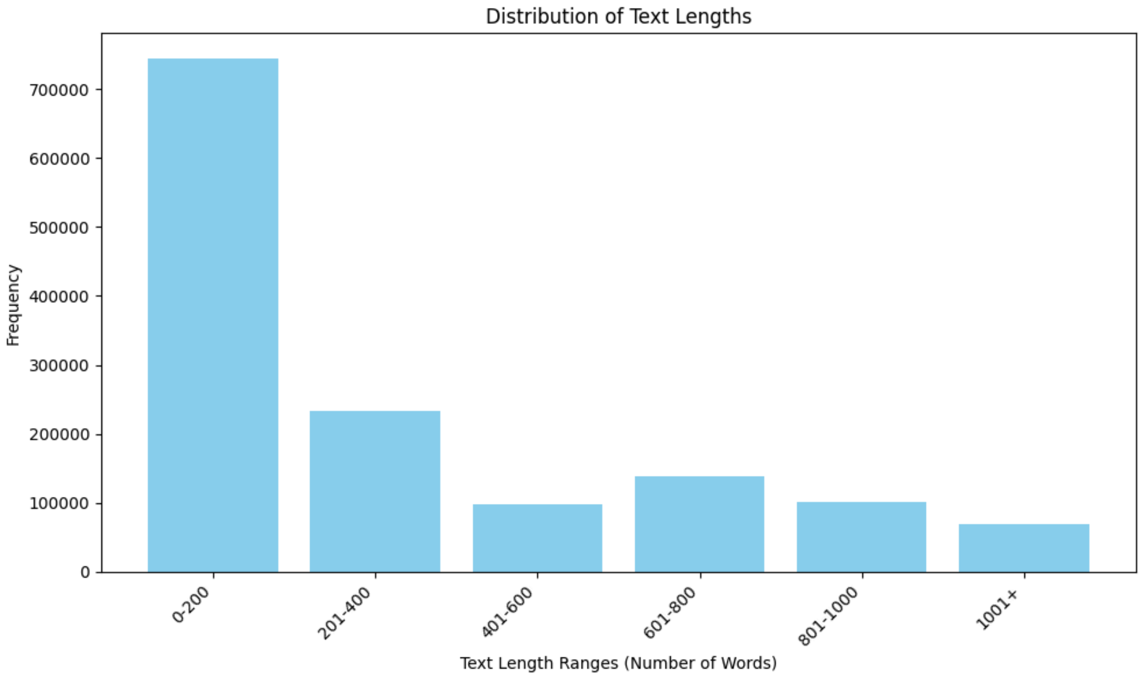


Figure 1.6 Dataset text lengths

Gaggar’s dataset has limited domain diversity, particularly in its AI-generated text, as it primarily focuses on a narrow set of contexts derived from specific sources. Additionally, the dataset only includes text samples ranging from 10 to 200 words, which further restricts its applicability to longer or shorter forms of text often encountered in real-world applications. This constraint limits the dataset’s ability to fully capture the varied linguistic styles and patterns present in broader contexts. This study also expands the scope by incorporating GAN-generated synthetic AI text to augment the dataset, offering greater diversity and adaptability in machine-generated text compared to static rephrased samples from GPT-3.5 Turbo. The GAN model hyperparameters are fine-tuned on their dataset for fair comparison.

H. Significance of the Study

With the advancement of technology and the growing interest in artificial intelligence, one of the most fascinating achievements in machine learning is the ability of artificially intelligent algorithms to generate texts. Automatically generated texts“ commonly referred to as “artificial intelligence-generated texts", "artificial texts”, or “generative texts”, represent a revolution in the way computers can co-create and communicate with humans.

However, along with the development of these technologies, challenges arise, particularly concerning authenticity and credibility. As text generation technologies become increasingly sophisticated, it becomes easier to manipulate information, create false content, and even conduct disinformation campaigns, which may motivate conflict between individuals. The consequences of this phenomenon can lead to a breach of societal trust, severe implications for reliable sources of information, and a deterioration in the quality of public discourse.

For this reason, it is essential to develop practical tools and strategies that enable us to detect and safeguard against misinformation successfully. Researchers and organizations worldwide are collaborating to develop detection and verification technologies that will help maintain the authenticity of information and limit the impact of false machine-generated text.

The design and development of the GAN-Enhanced AI-generated Text Detection, is an innovative tool for identifying texts generated by artificial intelligence. It can be used immediately as it directly solves a current issue with broad concerns regarding AI-generated text. This study contributes to the development of the field of artificial intelligence. Also, it presents all stakeholders with the opportunity to verify the origin of the texts, thus enhancing the transparency and reliability of the digital content.

The study's results may be most beneficial to the following groups:

- **AI Advocates** - In the emerging field of AI text generation, there will be a higher need for oversight in digital content creation. The system would thus become a valuable tool for these groups, providing a dependable way to understand the origin of digital text and ensure ethical compliance.
- **Academic and Industry Organizations** - By distinguishing between AI-generated and human-made text, the model is an essential tool for ensuring academic integrity. It has the potential to aid in the detection of plagiarism and the preservation of original scholarly work, thereby upholding the standards of academic excellence.
- **Future Researchers** - This study's results in detecting AI-generated text and human-written text with the proposed models could serve as a baseline for improvement for future researchers of the computer science community.

I. Definition of Terms

Domain. It represents the contextual or thematic area to which a text belongs, helping classify it into predefined categories based on its semantic content.

Embedding. Represents the semantic meaning of the tokens (words, or characters) from text data, enabling the model to perform tasks like classification and text generation effectively.

Generative Adversarial Network (GAN). A method for generative modeling that utilizes deep learning techniques, particularly convolutional neural networks (CNN).

Generative Pre-trained Transformer (GPT). Uses techniques from deep learning to generate natural language text, such as articles, narratives, or conversations, that resembles those created by humans.

Machine Learning (ML). A branch of artificial intelligence that utilizes statistical methods to enable computer systems to acquire knowledge from data without the need for explicit programming.

Natural Language Processing (NLP). A branch of artificial intelligence (AI) that focuses on training computers to understand, generate, and modify human language.

RoBERTa. A natural language processing system, BERT, has been enhanced for pretraining.

Text. A natural language data in the form of written words, sentences, or paragraphs.

Threshold. A set numeric value in classification models at which the predicted probability (or score) tips from one class to another. The placement of this cutoff value, sets the model's **"decision boundary"** – determining when it should classify an example as one category or the other.

Chapter II : Review of Related Literature and Studies

A. Generative Adversarial Network (GAN)

The field of AI text generation has shown significant growth with the use of Generative Adversarial Networks (GANs). These networks have entirely transformed our approach to creating and identifying artificial text. The basic theory of GANs revolves around the dialectical interaction between two neural networks—the generator, responsible for generating data, and the discriminator, tasked with evaluating the generated data.

Croce et al. (2021) demonstrated the use of Generative Adversarial Networks (GANs) to improve text classification accuracy when only a small number of labeled examples are available. The study showed the model's ability to effectively utilize unlabeled data to enhance performance in various text-related tasks. Semi-supervised learning is essential to their study, as it attempts to distinguish AI-generated text from human-produced language utilizing a GAN-based approach.

In bot detection, Najari, Salehi, and Farahbakhsh (2022) introduced GANBOT, a GAN-based framework that effectively detects bots on social media platforms. Their study utilized a semi-supervised learning model that uses GAN to analyze and predict social bots' behavior more effectively. The framework of their model uses a generative model with a classification model with a shared Long Short-Term Memory (LSTM) layer. The study showed a significant increase in the probability of bot detection in their model. The GANBOT model improved the true positive rate, the percentage of actual bots that were efficiently detected. Their study emphasized the framework's superior performance over traditional Contextual LSTM methods, particularly highlighting its effectiveness in reducing false positives and increasing true detection rates.

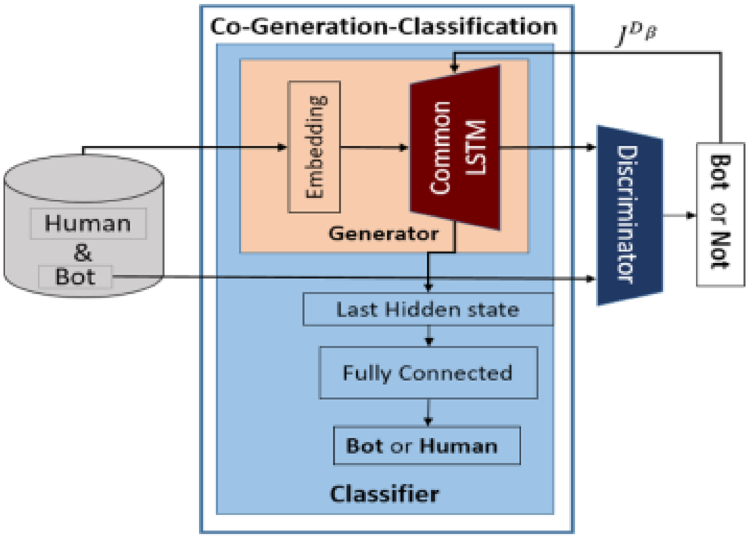


Figure 2.1 The proposed GANBOT framework (Najari, Salehi, & Farahbakhsh, 2022)

In another study conducted by Aghakhani et al. (2018) on detecting deceptive reviews using GANs using their proposed FakeGAN model, they decided to use two discriminators instead of the traditional GAN, which only uses one - to distinguish between real and generated samples. The two discriminators in the study have the following tasks: The first discriminator, labeled D, distinguishes between deceptive and truthful reviews, while the second discriminator, labeled D’, distinguishes deceptive reviews between those generated from the generator and those from the training dataset. Their idea behind using two discriminators is to be able to come up with a stronger generator model. It was mentioned that during the adversarial learning phase, the generator only gets rewarded from one discriminator. Hence, there is a chance GAN may result in mode collapse as it tries to learn two different distributions (the truthful and deceptive reviews). Mode Collapse is a typical training issue in GANs in terms of stability, wherein the generator does not get to capture the entire diversity of the training data and only gets to produce a limited set of similar outputs. In other words, it is where the generator "collapses" after a few patterns/modes rather than being able to generate diverse and realistic samples. Thus, their solution to this was to have two discriminators. Furthermore, in their model, using two discriminators makes discriminator 2 train the generator to make better deceptive reviews and simultaneously trains discriminator 1 to be a better discriminator.

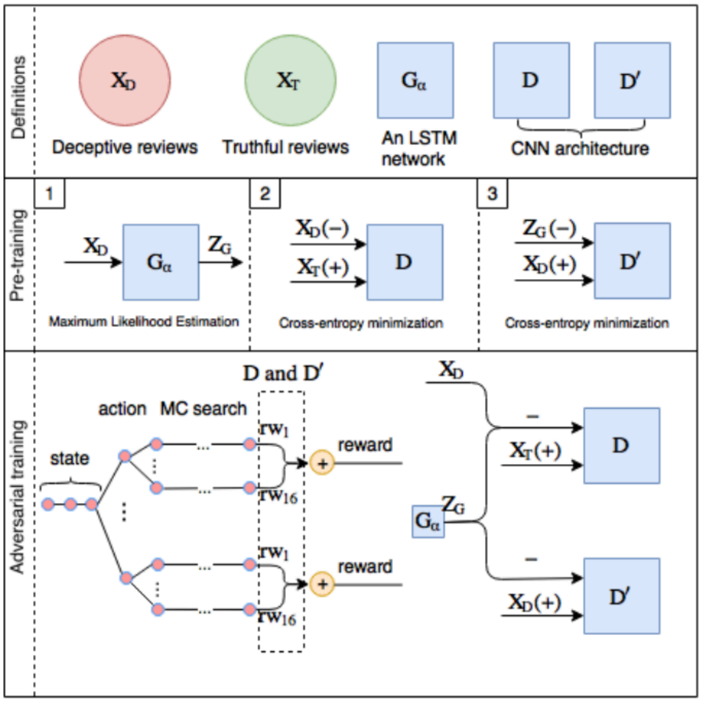


Figure 2.2 The overview of the FakeGAN model for deceptive reviews detection (Aghakhani, Machiry, Nilizadeh, Kruegel & Vigna, 2018)

Figure 2.2 Shows the overview of the FakeGAN model from their study. Using the feedback from both D and D’, the generator would try to fool both discriminators by generating reviews that look deceptive to D’ and truthful to D (not generated by the generator/came from the dataset). MLE, Maximum Likelihood Estimation, was used to train the generator on deceptive reviews from the training dataset. The Minimizing Cross-Entropy technique was used to pre-train the discriminators. Their generator was defined as a policy model, a stochastic model, in reinforcement learning. It was also trained using policy gradient and Monte Carlo searches on the expected end reward from the two discriminators D and D’.

B. Generative Pretrained Transformer (GPT)

The study of Gaggar, Bhagchandani, and Oza (2023) demonstrated in the field of Generative Pretrained Transformer (GPT) by focusing on text produced by large language models (LLMs), like OpenAI's ChatGPT, which are capable of generating highly realistic text.

To start with, the authors effectively created a well constructed labeled dataset encompassing texts originated by both humans and ChatGPT (GPT3.5 turbo) from five distinct sources. The sentences in the dataset exhibit a varied length range, spanning from 10 to 200 words. Additionally, they have designed and trained multiple classifiers to differentiate between texts generated by humans and ChatGPT.

The dataset used in their study is a combination of five different sources of human-generated text, including Twitter Sentiment, Football Commentary, Project

Gutenberg, PubMedQA, and SQuAD datasets. The purpose of using these datasets is to improve the model's ability to understand and analyze both short-form and long-form text and complex medical jargon. The merged human-generated text resulted in 2,534,498 sentences, which were grouped into categories of 10-200 words with 5-word increments. Normalization was performed to ensure 16% of sentences in each category, resulting in a final dataset of 400,015 sentences. To generate machine text, they rephrased the selected sentences using the GPT-3.5-turbo chatgptAPI from OpenAI, with a prompt to ensure that the length of the rephrased sentence is the same as the original sentence. The total number of rows in our dataset is 800,030, consisting of both human and machine generated text, with sentences ranging from 10 to 200 words in length. They truncated sentences longer than 200 words and removed those with fewer than 10 words.

Their RoBERTa-base model, enhanced with a (FC+sigmoid) layer, proves effective in our binary classification, nearing state-of-the-art performance observed in similar studies. Its adeptness in capturing intricate linguistic nuances is countered by its larger size (125 million parameters), demanding more training and inference resources. While excelling in accuracy, its computational complexity introduces challenges in efficiency when compared to the lightweight SVM model. This highlights the nuanced trade-offs between model sophistication, performance, and resource requirements within the context of our classification task.gthsof transformer-based language models while tailoring them to the specific demands of our binary classification task. For their experimental approach, they organized their datasets into distinct training, testing, and validation sets, each comprising sentences ranging in length from 10 to 200 words. To ensure a granular examination of model performance across different sentence lengths, they further divided the data within each set into specific ranges, such as 10-14 words, 15-19 words, and so on.

For their experimental results, it was shown that as sentence lengths increase, RoBERTa-base and RoBERTa-large models consistently outperform SVM, showcasing robustness in capturing nuanced patterns. In their study, they utilized fine-tuning RoBERTa on their dataset that includes both human-written and AI-generated text. The fine-tuning process is critical as it adapts RoBERTa's pre-trained capabilities to the specific patterns that characterizes AI-generated text. The utilization of RoBERTa exemplifies its capacity to handle complex, nuanced linguistic text, which helps in achieving more accurate performance results. The comparative analysis of RoBERTa-base and RoBERTa-large in their study provides insights into the trade-offs, where RoBERTa-base is generally more computationally efficient, and RoBERTa-large offers higher accuracy and a deeper understanding of the AI-generated or human-written text. As discussed in their study with RoBERTa, larger and more complex models like RoBERTa-large often provide better detection efficacy due to their deeper and more nuanced understanding of language. However, they also tend to be less computationally efficient because they require more computational power and longer processing times due to their size and complexity. The implications of their research extend into digital forensics and AI ethics, highlighting the growing need for sophisticated detection mechanisms that can keep pace with advancements in machine learning and AI-generated text.

C. Machine Learning (ML)

The use of machine learning applications is expanding in many areas. The study conducted by Prova (2024) presents an approach utilizing various machine-learning methods to distinguish between human-written and AI-generated texts. The study revolves around applying machine learning algorithms such as XGB Classifier, Support Vector Machine (SVM), and BERT architecture deep learning models, focusing on enhancing the accuracy of AI-generated text detection.

The study emphasizes the practical implications of utilizing advanced machine learning models to reduce the effect of inaccurate data. Their model showed that while traditional machine learning models like XGB Classifier and SVM showed considerable efficacy, with accuracies of 84% and 81%, respectively, BERT outperformed them significantly by achieving a 93% accuracy rate. This superior performance of BERT can be attributed to its deep learning framework, which excels in capturing the contextual difference that distinguishes AI-generated text from human-written text.

In another study by Islam et. al. (2023) presents machine-learning approaches to distinguish between human-generated and ChatGPT-generated text. The study utilizes a diverse range of machine learning and deep learning algorithms, including Logistic Regression, Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbors, Random Forest, AdaBoost, Bagging Classifier, Gradient Boosting, Multilayer Perceptron, Long Short-Term Memory (LSTM), and an Extremely Randomized Trees Classifier (ERTC). The use of ERTC, a variant of ensemble learning that utilizes randomness in building trees, showcased a high performance, achieving an accuracy of 77%. These multiple algorithms allow for a comprehensive comparative analysis to determine the most effective method for correctly classifying texts.

D. Natural Language Processing (NLP)

Natural Language Processing (NLP) allows machines to understand and interpret human language meaningfully. As AI has advanced, this field has expanded quickly. In particular, the development of transformer models, which have raised the requirements for various NLP tasks, has contributed to this growth.

The widespread use of digital text has contributed to development of advanced methods for handling and interpreting massive amounts of textual data. The creation of transformer architectures, such as BERT and GPT, which enabled sequential NLP tasks and gave the development of large language models that could replicate human behavior, was the turning point in natural language processing.

Several methods are available in recent research for identifying writings produced by AI. GLTR, a tool for statistical identification and display of created text, was studied by Gehrmann, Strobelt, and Rush (2019). It can be used to find text that differs from common patterns written by humans. According to their research, the GLTR annotation method raises the human detection rate of fake text from 54% to 72% without any prior training. This is very similar to our research's objectives, which are to identify patterns that might take time to be obvious to differentiate text produced by AI from that created by humans.

In the study of Gaggar et al. (2023), the role of NLP in enhancing the ability to discern between human-written and ai-generated text is highlighted. This is relevant to our proposed research because we will also be using NLP methods, to refine and optimize our proposed model’s performance. Gaggar et. al’s study used a comprehensive approach where they utilized deep analysis of text patterns and styles, which are characteristics of the output from modern language models like GPT-3.5. In their study’s methodology, NLP was not only used for preprocessing but it was also integrated in their model’s training and testing phase.

Our proposed research uses these advancements to address the problems presented by texts produced by AI that are almost identical to those written by humans. While most previous research has concentrated on using transformer models to generate or classify texts, the group’s proposed research focuses on using these advances in NLP to verify the identity of the text's source, be it human or artificial intelligence.

E. RoBERTa

In the study of Oghaz, Dhame, Singaram, & Saheer (2023), they used a custom RoBERTa-based model. Their study aimed to address the challenges posed by AI in generating authentic-looking text that could be used for malicious purposes like spreading misinformation or cheating in academic settings. Their RoBERTa-based model, which was fine-tuned, demonstrated exceptional performance, achieving an F1-score of 0.992 and an accuracy of 0.991. These results were better than other models like DistilBERT, which also performed well but slightly lower with an F1-score of 0.988.

Another study conducted by Wang, Cheng, and Yu (2024) used both BERT and fine-tuned RoBERTa to detect AI-generated news by GPT models, such as ChatGPT, The approach taken by Wang et al. involved fine-tuning RoBERTa on large datasets that included instances of both human-made and AI-generated. This method allowed the model to learn and identify subtle linguistic features that distinguish AI-generated content from human-written articles. Their findings demonstrated that models like RoBERTa could effectively apply to real-time detection.

The current studies confirm the effectiveness of RoBERTa in tasks related to text authenticity. Nonetheless, the group's approach lies in the integration with GAN, aiming to improve the model's ability to handle the increasingly sophisticated output of modern AI text generators. This integration aims to enhance the model's ability to detect by categorizing text and identifying significant patterns that may go unnoticed by traditional techniques.

In another study by Chen & Liu (2023), they introduced STADEE (Statistics-Deep Detection of Machine Generated Text), a method to enhance the detection of machine-generated text, mainly that came from advanced language models like GPT-3 and GPT-4. The study utilized a fine-tuning approach where RoBERTa was explicitly trained to detect text generated through nucleus sampling. The RoBERTa was trained using a batch size of 48, a learning rate 5×e−5, the AdamW optimizer, and fine-tuned for 2 epochs. RoBERTa has an F1-score of 97.96 and 98.37 for validation and testing sets, respectively, surpassing the performance of other models. In comparison, GLTR achieves scores of 78.19 and 77.77, while STADEE achieves scores of 87.65 and 87.05. The study found that RoBERTa, when fine-tuned, was notably effective against texts generated by large and smaller models, showing its adaptability to various text generation methods.

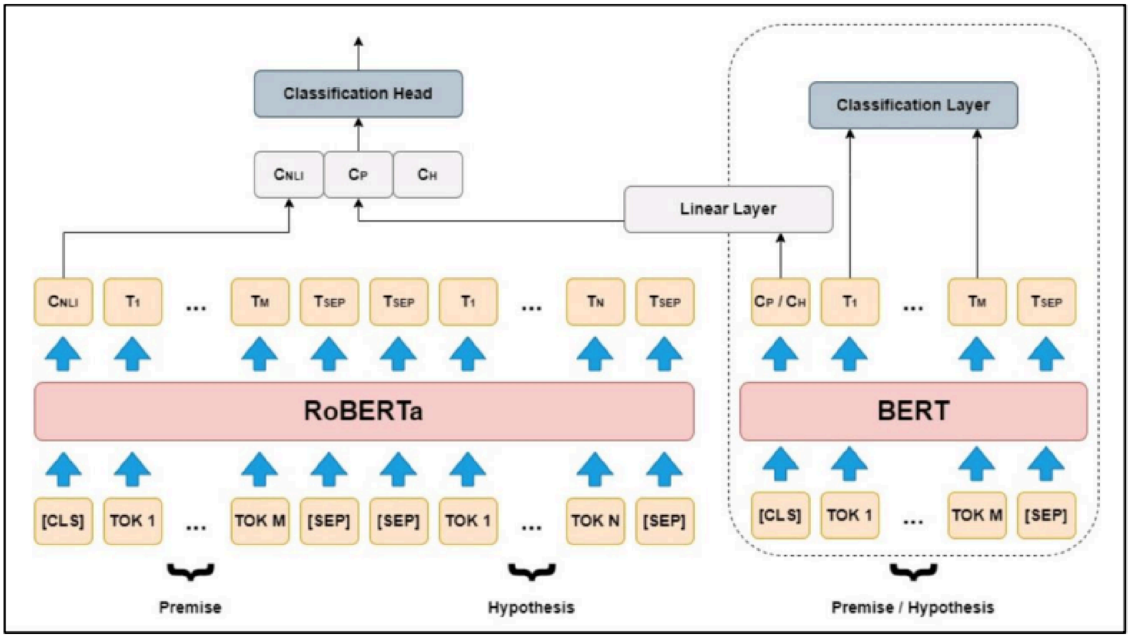


Figure 2.3 RoBERTa Architecture

F. Synthesis

Table 2.1 Synthesis of Related Studies

Title	Author & Year Published	Approach	Conclusion	Findings
RoBERTa				
Implementing BERT and fine-tuned RobertA to detect AI-generated news by ChatGPT	Zecong Wang, Jiaxi Cheng, Chen Cui, & Chenhao Yu (2024)	Used BERT and Fine-tuned RoBERTa on large datasets	Demonstrated that the fine-tuned RoBERTa model effectively distinguished between human-written and AI-generated. Shows the effectiveness of using advanced deep learning models such as RoBERTa	The precision of 98% with the fine-tuned RoBERTa model.
Detection and Classification of ChatGPT Generated Contents Using Deep Transformer Models	Mahdi Maktab Dar Oghaz, Kshipra Dhame, Gayathri Singaram, & Lakshmi Babu Saheer (2023)	Used several machine learning and deep learning models, including: Multinomial Naive Bayes, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Random Forest, Bidirectional LSTM (BiLSTM), DistilBERT, & RoBERTa	Deep learning models, particularly those based on transformer architecture like RoBERTa and DistilBERT, were more effective than traditional machine learning models in the classification tasks. These models efficiently handled the complexity of the language patterns used by AI text generators.	RoBERTa-based model's F1-score: 0.992 and accuracy: 0.991. DistilBERT's F1-score: 0.986
STADEE: STAtistics-based DEEp Detection	Zheng Chen, & Humming Liu (2023)	Used a fine-tuned RoBERTa and was trained to detect	Indicated that a detection model trained on text	97.96 and 98.37 F1 scores for RoBERTa.

of Machine-Generated Text		text through nucleus sampling	generated by larger models (such as those employing nucleus sampling) could effectively detect text produced by smaller models.	
RoBERTa: A Robustly Optimized BERT Pretraining Approach	Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, & Veselin Stoyanov (2019)	Trained the model for longer periods than BERT, which had been significantly undertrained. Increased batch size during training helps stabilize the learning and leverage modern parallel computing resources more effectively.	Concluded that many of the improvements claimed by newer models over BERT could actually be achieved by adjusting the training procedure and not necessarily by modifying the architecture or introducing new training objectives	RoBERTa with 88.5 on General Language Understanding Evaluation (GLUE) when trained with optimized procedures, while XLNeT got 88.4.
Generative Adversarial Network (GAN)				
Exploring Ensemble Models and GAN-Based Approaches for Automated Detection of Machine-Generated Text	Surbhi Sharma (2024)	Used the GAN-RoBERTa framework, integrating Generative Adversarial Networks (GANs) and RoBERTa, utilizing labeled and unlabeled datasets	GAN-RoBERTa demonstrated notable improvements in detecting machine-generated text, especially when leveraging unlabeled data for adversarial training.	GAN-RoBERTa model achieved an accuracy of 73.6% on mixed labeled and unlabeled data. F1 score of 83.26% for distinguishing between human and machine-generated text.
GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of	Danilo Croce, Giuseppe Castellucci, & Roberto Basili (2020)	Used GAN-BERT, combining BERT fine-tuning with unlabeled data in a generative adversarial setting.	Suggested use of adversarial training for semi-supervised learning transformer-based architectures.	BERT accuracy: 22.2% GAN-BERT accuracy: 30.4%

Labeled Examples				
GANBOT: a GAN-based framework for social bot detection	Shaghayegh Najari, Mostafa Salehi, & Reza Farahbakhsh (2022)	Used GAN with Contextual Long Short-Term Memory (LSTM) and GloVe (Global Vectors of World Representation	Concluded that the GANBOT framework significantly improved the detection of social bots, especially by utilizing a GAN method to generate more detailed samples of bot behavior.	The GANBOT model enhanced the true-positive rate and emphasized their framework's superior performance over traditional Contextual LSTM methods, reducing false-positive and increasing true detection rates.
RADAR:Robust AI-Text Detection via Adversarial Learning	Xiaomeng Hu, Pin-Yu Chen, & Tsung-Yi Ho (2023)	Created synthetic language using a Generative Adversarial Network (GAN) model with datasets like Xsum, TOEFEL, Reddit WritingPrompts, SQuAD, and WP.	Emphasized that RADAR not only outperforms existing detection models but also excels in more challenging conditions where the texts have been modified to resemble human writing more closely.	OpenAI's RoBERTa model outperformed RADAR's 85.6% AUROC score with 90%. When applied with a seen or unseen paraphraser, RADAR's model got 92.2% and 85.7% AUROC scores, compared to OpenAI's RoBERTa model's 80.6% and 62.7%.
Detecting Deceptive Reviews using Generative Adversarial Networks	Hojjat Aghakhani, Aravind Machiry, Shirin Nilizadeh, Christopher Kruegel, & Giovanni Vigna (2018)	Involves two discriminators, labeled D and D', working alongside a generator.	Demonstrate that GANs, particularly with the novel dual-discriminator design, can effectively adapt to the challenges of	FakeGAN achieved an accuracy: of 89.1%.

			text classification without requiring extensive labeled datasets.	
AI-generated text detection				
Machine-Generated Text Detection using Deep Learning	Raghav Gaggar, Ashish Bhagchandani, & Harsh Oza (2023)	Used Support Vector Machines (SVM), RoBERTa-base, and RoBERTa-large	Highlights the importance of ongoing advancements in detection technologies to keep pace with the evolving complexity of language models	Test AUC-ROC scores: RoBERTa-large's: 95.53% RoBERTa-base: 95.24% SVM: 84.81%
Detecting AI Generated Text Based on NLP and Machine Learning Approaches	Nuzhat Noor Islam Prova (2024)	Uses XGB Classifier, Support Vector Machine (SVM), and BERT architecture deep learning models	The BERT model, due to its deep learning architecture and ability to process large amounts of training data, was highly effective in distinguishing AI-generated text from human-written text.	XGB Classifier and SVM's accuracy: 84% and 81% respectively, BERT's accuracy: 93%
Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning	Niful Islam, Debopom Sutradhar, HumairaNoor , Jarin Tasnim Raya, Monowara Tabassum Maisha, & Dewan Md Farid (2023)	Uses machine learning and deep learning algorithms, including Logistic Regression, Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbors, Random Forest, AdaBoost, Bagging Classifier, Gradient Boosting,	Machine learning models, particularly the Extremely Randomized Trees Classifier (ERTC), are highly effective in distinguishing between AI-generated and human-generated texts.	Vectorizing sentences using TF-IDF's accuracy: 77%

		Multilayer Perceptron, Long Short-Term Memory (LSTM), and an Extremely Randomized Trees Classifier (ERTC).		
GLTR: Statistical Detection and Visualization of Generated Text	Sebastian Gehrmann, Hendrik Strobelt, Alexander M. Rush (2019)	Using their model GLTR, the methods they used are simple yet effective statistical detection methods to aid in distinguishing AI-generated text from human-written text.	Found that their model's effectiveness of integrating simple statistical methods within an interactive tool to enhance the detection of AI-generated text	GLTR improves the human detection rate of fake text from 54% to over 72% accuracy.
DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature	Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, & Chelsea Finn (2023)	Utilized the model, DetectGPT and generated minor rewrites (perturbations) of a text passage using a different model such as T5	Demonstrated superior performance in detecting machine-generated text, notably improving the Area Under the Receiver Operating Characteristic (AUROC) scores compared to existing zero-shot detection methods	Increased the AUROC score from 0.81 to 0.95 generated by GPT-NeoX, with a 20 billion parameter model.

Chapter III: Research Design and Methodology

A. Hypothesis

The researchers implemented an application of Generative Adversarial Network (GAN) for text classification, which differs from the study of Gagger et al. (2023), wherein they only used a simple frozen parameter variation for their RoBERTa model. With that, the researchers tested the following hypotheses:

> **Ho:** The implementation of a model that used GAN does not have a significant difference compared to a model that used RoBERTa.
>
> **Ha:** The implementation of a model that used GAN does have a significant difference compared to a model that used RoBERTa.

Assumptions

The research is under the following assumptions:

1. All the dataset samples are correctly labeled as human or AI.
2. The dataset comprises only English text sentences of various lengths.
3. The dataset samples cover a wide range of domains.

B. Research Methods

This study conducted experimental research objectively and controlled so that specific conclusions will be drawn from the hypothesis statements. This study aimed to identify the cause-and-effect relationships between the independent variable, which in this case is the GAN model, with the dependent variable, which is the model's overall

AI-generated text detection performance in terms of accuracy, precision, recall, and F1 scores.

This study based its framework on the RoBERTa-base model of the study conducted by Gaggar et al. (2023) on Machine-Generated Text Detection using Deep Learning. The study above would serve as the basis for comparison for this research wherein the RoBERTa-base model, augmented with an FC (fully connected) layer and sigmoid activation function, was used for binary classification. Our proposed model incorporated Generative Adversarial Networks in comparison to the RoBERTa-base model used in the basis study. The performance is evaluated by evaluating the performance of a model using a GAN model and to the model of Gaggar et al. (2023).

C. Research Design

The proposed research design integrates a comprehensive computational approach to distinguish between AI-generated and human-written texts. The study leverages a Generative Adversarial Network (GAN) to train a discriminator capable of effectively handling diverse linguistic inputs.

To begin, this study referred to academic literature from sources such as arXiv, Worcester Polytechnic Institute Digital, ACL Anthology, and ResearchGate. The dataset used in this research is sourced entirely from Gaggar's dataset, which includes five distinct domains of human-generated text: Twitter Sentiment, Football Commentary, Project Gutenberg, PubMedQA, and SQuAD datasets. Additionally, it features AI-generated text samples paraphrased by ChatGPT, ensuring a balanced representation of human-written and AI-generated text for classification.

The research methodology follows three key phases. Initially, the text data undergoes rigorous pre-processing, including cleaning, and tokenization, to prepare it for training. In the second phase, the GAN model is trained to optimize the discriminator for classifying text as either human or AI-generated. This involves using both real and synthetic samples to improve the discriminator's ability to generalize. Finally, the model's performance is evaluated on the testing set using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

The strength of this research design lies in its focus on a GAN-based approach compared to Gaggar’s RoBERTa model to enhance classification performance. By focusing on training the GAN discriminator directly on the dataset, the model is exposed to diverse linguistic scenarios within the real human and AI text samples, improving its robustness and accuracy. Unlike traditional single-model systems, this method leverages fine-tuning of the discriminator to optimize its classification performance without relying on synthetic AI text augmentation. The entire process is carefully documented to enable replication by other researchers and ensure consistent results across different experimental setups. This approach provides valuable insights into differentiating between AI-generated and human-written texts while highlighting the impact of directly training the GAN discriminator on the dataset.

D. Research Instruments

a. Hardware

The researchers used an AMD Ryzen7-7735HS overclocked to 4.7GHz clock speed, an NVIDIA GeForce RTX 4050 with 6GB GDDR6 VRAM, and 16GB DDR5 RAM running at 2420MHz.

b. Software

Python will be used to code the program while implementing testing parameters, and the code for testing will be run using Google Colab Pro to make up for limited memory resources during runtime. The researchers utilized external Python libraries such as NumPy, Pandas, Scikit-learn, HuggingFace/Transformer, as well as Pytorch. The dataset used in this study was obtained from HuggingFace.

c. People

The people involved in this study were mainly the researchers together with the guidance of both our thesis adviser and thesis coordinator.

E. Data Acquisition and Preparation

The dataset used in this study is sourced from Gaggar's dataset, which is a combination of five different sources of human-generated text, including Twitter Sentiment, Football Commentary, Project Gutenberg, PubMedQA, and SQuAD datasets. This diverse dataset is designed to improve the model's ability to understand and analyze both short-form and long-form text, as well as complex medical jargon. Additionally, the dataset includes AI-generated text samples paraphrased by ChatGPT, ensuring a balanced representation of human-written and AI-generated text for classification tasks.

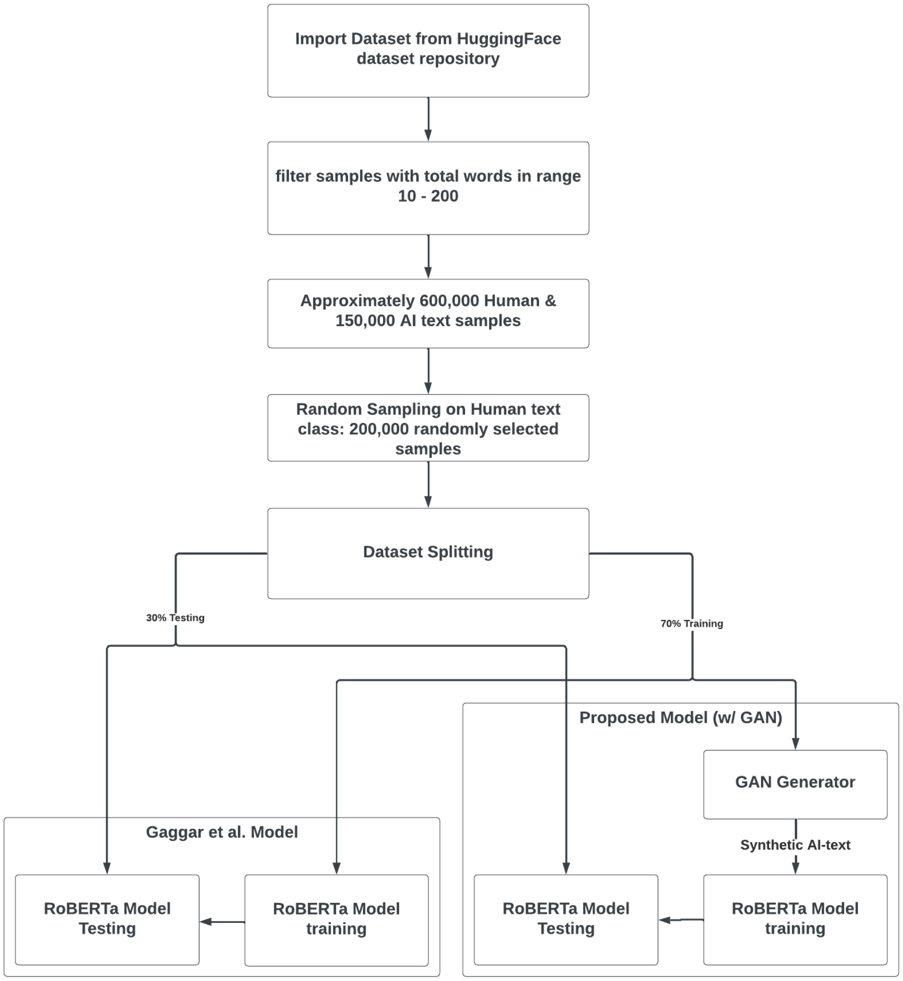

Import Dataset from HuggingFace dataset repository
filter samples with total words in range 10 - 200
Approximately 600,000 Human & 150,000 AI text samples
Random Sampling on Human text class: 200,000 randomly selected samples
Dataset Splitting
30% Testing
70% Training
Proposed Model (w/ GAN)
GAN Generator
Synthetic AI-text
Gaggar et al. Model
RoBERTa Model Testing
RoBERTa Model training
RoBERTa Model Testing
RoBERTa Model training

Figure 3.1 Visualization of data preparation

The dataset consists of human-written and AI-generated text samples sourced from Gaggar et al. 's dataset, containing 400k AI and 380k human text samples. To ensure consistency and fairness, the dataset undergoes preprocessing steps, which include the removal of special characters, extra spaces. After cleaning, domain classification is conducted using ChatGPT to categorize the text into relevant domains. The processed dataset is tokenized, and an 80-10-10 split is applied for training, validation, and testing, respectively.

Table 3.1 Example generating domain-classified datasets

Prompt	ChatGPT 4o Response
Can you classify then add a column that shows what kind of domain/category it is part of: The categories are limited to: - Lifestyle - Arts/Culture - Science/Technology - Health - Politics - Business/Economics - Sports - Education I want you to give me the modified csv for part 1.	datasetbatch_withdomains.csv

Table 3.2 Dataset with Classified Domains count

Domain / Category	Count
Lifestyle	233968

Arts/Culture	177435
Science/Technology	35396
Health	26463
Politics	20273
Sports	13789
Business/Economics	13651
Education	11594
TOTAL	**536388**

To ensure our dataset was categorized into relevant domains for further analysis, we utilized ChatGPT 4o's ability to classify text samples. Given the limitations in file size for processing, the dataset was divided into multiple CSV batches. Each batch was processed individually, with ChatGPT 4o appending a new column labeled "Category," representing the assigned domain. The predefined categories included Lifestyle, Arts/Culture, Science/Technology, Health, Politics, Business/Economics, Sports, and Education. Once all batches were processed, they were combined to create a comprehensive dataset with domain classifications.

For the GAN-based system, the discriminator is trained using both human and AI text samples to classify the source of the text. The generator uses noise and embeddings from Human text and AI text samples to produce synthetic text. While the generator aids in augmenting the training process, the discriminator is the primary focus, as it is fine-tuned and evaluated for its classification performance. The model’s performance is iteratively validated using metrics like accuracy and AUC-ROC, and hyperparameters are adjusted based on the validation set outcomes. The final discriminator model is then tested on unseen test data to assess its ability to classify human and AI text, providing insights into the effectiveness of the GAN training process.

F. Statistical Treatment of the Data

The metrics, accuracy, precision, recall, f1-score, confusion matrix, and a statistical t-test were used to evaluate the overall performance of the model.

Accuracy will be crucial for assessing the overall effectiveness of the GAN-enhanced RoBERTa model in correctly identifying the text's origin. It represents the proportion of true results (both true positives and true negatives) among the total number of samples examined. The formula for the accuracy metrics is as follows:

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{Total\ Number\ of\ Samples}$$

Figure 3.2 Accuracy formula

F1-Score is a reliable measure when dealing with imbalanced datasets that might occur if one class of texts (AI-generated or human-made) dominates over the other. The F1-score is the harmonic mean of precision and recall, balancing the model's sensitivity (recall) and its ability to remain error-free (precision). The formula for the f1-score metrics is as follows:

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Figure 3.3 F1 score formula

Precision and Recall are implicitly important through the F1-score. Precision (also called positive predictive value) measures the accuracy of positive predictions. Recall (sensitivity) measures the ability of a model to find all the relevant cases (all positive samples). The formula for the precision metrics and recall metrics are as follows:

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$$

Figure 3.4 Precision and Recall formulas

The **Confusion Matrix** offers detailed insight into the performance of the model beyond what single metric summaries like accuracy can provide. This can identify whether the model suffers from bias (overfitting) or variance (underfitting) and adjust the training strategies accordingly.

Table 3.3 Confusion matrix table

	Predicted AI-Generated	**Predicted Human-written**
Actual AI-Generated	TP (True Positive)	FN (False Negative)

Actual Human-written	FP (False Positive)	TN (True Negative)

Where:

TP (True Positive): Number of texts correctly identified as AI-generated.
TN (True Negatives): Number of texts correctly identified as human-written.
FP (False Positives): Number of human-written texts incorrectly identified as AI-generated.
FN (False Negatives): Number of AI-generated texts incorrectly identified as human-written.

The **AUC-ROC** Score is a metric for evaluating the performance of the GAN-RoBERTa model, which aims to distinguish between human-written and AI-generated texts. Given the binary nature of this classification task (human vs. AI), the AUC ROC score provides a comprehensive measure of the model's ability to correctly classify texts across various thresholds.

$$AUC\ ROC = \int_{0}^{1} TPR(t)\ dFPR(t)$$

Figure 3.5 AUC-ROC score formula

Where:

$TPR(t)$ is theTrue Positive Rate at threshold t.

$FPR(t)$ is the False Positive Rate at threshold t.

$dFPR(t)$ represents the differential change in the False Positive Rate.

The **Self-BLEU** Score is a metric used to measure the diversity of generated text samples by comparing individual outputs against each other. It is adapted from the traditional BLEU score but evaluates how similar a generated sample is to others in the same dataset, with higher scores indicating lower diversity. This metric is particularly useful in assessing whether a GAN is producing repetitive or varied text. The formula for SELF-BLEU is as follows:

$$Self - BLEU = \frac{1}{N}\sum_{i=1}^{N} BLEU(G_i, G \setminus G_i)$$

Figure 3.6 Self-Bleu score formula

Where:

N is the number of generated samples..

G_i is the i-th generated sample..

$G \setminus G_i$ represents the set of all generated samples except G_i.

Negative Log-Likelihood (NLL) is a metric used to evaluate the quality of the generated text by assessing how well the generator models the true data distribution. A lower NLL score indicates a better fit between the generated samples and the target distribution, as it measures the average log probability of the true samples under the generated distribution. The formula for NLL is as follows:

$$NLL = -\frac{1}{N}\sum_{i=1}^{N} log\, P(y_i|x_i)$$

Figure 3.7 Negative Log-Likelihood formula

Where:

N is the number of generated samples.

y_i is the true class label (real of synthetic)

x_i is the input text sample.

$P(y_i|x_i)$ is the predicted probability of the true class label.

To test the hypothesis, the **Confidence Intervals (CI)** of the test accuracies of the RoBERTa model with GAN and without GAN are compared. If the confidence intervals overlap, then there is no significant difference. If they do not overlap, then there is a significant difference.

$$Confidence\ Interval\ =\ X \pm Z(\sigma/\sqrt{n})$$

Figure 3.8 Confidence Interval formula

Where:

X is the sample mean.

Z is the Z-score associated with the desired confidence level.

σ is the population standard deviation.

n is the sample size.

Chapter IV Presentation and Analysis of Data

A. System Architecture

The system architecture of this project outlines the processes and components involved in evaluating the GAN-based discriminator for distinguishing between human-written and AI-generated text. Starting with data preprocessing until GAN training, utilizing both human and AI text alongside noise to update the generator and discriminator. The trained discriminator is validated and tested to evaluate performance using metrics such as accuracy and AUC-ROC. Figure 4.1 Shows a high-level visualization that highlights how the data flows through each stage to achieve optimal classification results.

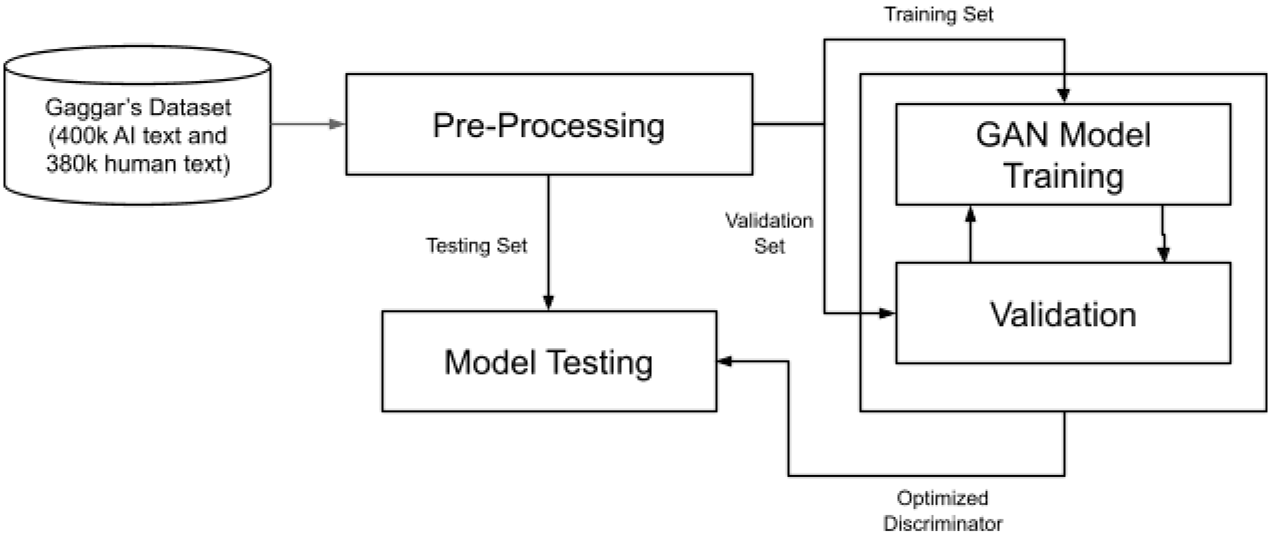
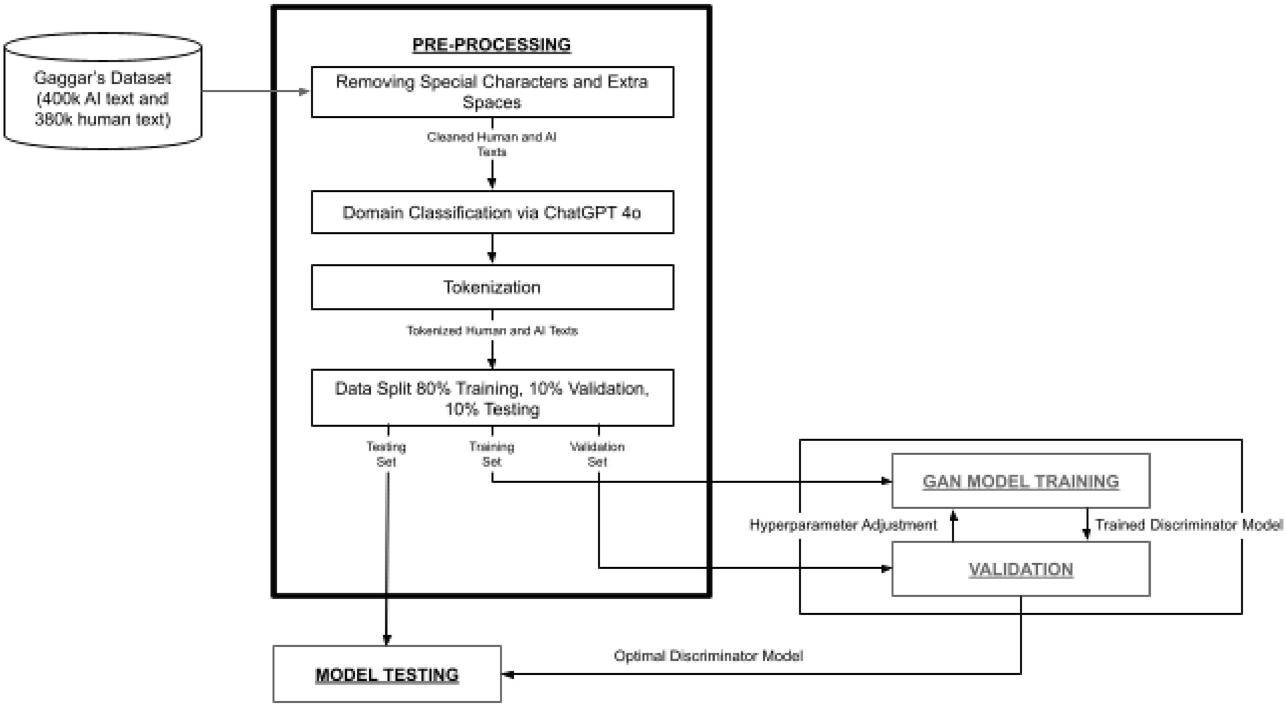


Figure 4.1: System Architecture (High Level)


Gaggar's Dataset
(400k AI text and
380k human text)
PRE-PROCESSING
Removing Special Characters and Extra Spaces
Cleaned Human and AI Texts
Domain Classification via ChatGPT 4o
Tokenization
Tokenized Human and AI Texts
Data Split 80% Training, 10% Validation, 10% Testing
Testing Set
Training Set
Validation Set
GAN MODEL TRAINING
Hyperparameter Adjustment
Trained Discriminator Model
VALIDATION
Optimal Discriminator Model
MODEL TESTING

(Figure 4.1a: System Architecture (Pre-processing

The system architecture begins with Gaggar's dataset, which includes 400k AI-generated text samples and 380k human-written text samples. During the pre-processing stage, the texts undergo cleaning to remove special characters and extra spaces. After cleaning, the texts are categorized into domains using ChatGPT 4.0. Once the domain classification is complete, the texts are tokenized to convert them into a format suitable for model training. The tokenized dataset is then split into three subsets: 80% for training, 10% for validation, and 10% for testing. These subsets serve as inputs for subsequent phases of the system.

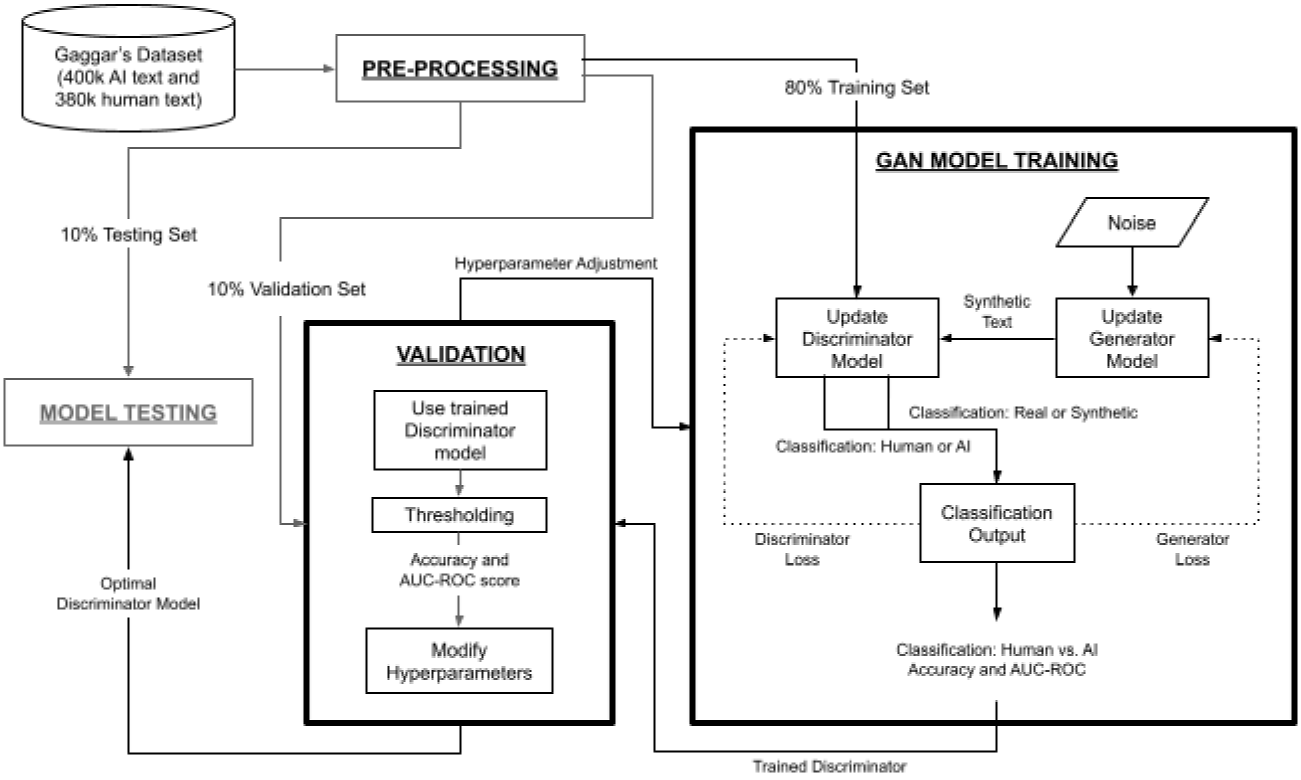

Gaggar's Dataset (400k AI text and 380k human text)
PRE-PROCESSING
80% Training Set
10% Testing Set
10% Validation Set
Hyperparameter Adjustment
GAN MODEL TRAINING
Noise
Update Discriminator Model
Synthetic Text
Update Generator Model
Classification: Real or Synthetic
Classification: Human or AI
Classification Output
Discriminator Loss
Generator Loss
Classification: Human vs. AI
Accuracy and AUC-ROC
Trained Discriminator
VALIDATION
Use trained Discriminator model
Thresholding
Accuracy and AUC-ROC score
Modify Hyperparameters
MODEL TESTING
Optimal Discriminator Model

Figure 4.1b: System Architecture (GAN Model Training and Validation)

In the GAN model training phase, a generator and discriminator are iteratively updated. The discriminator learns to classify text samples as either human-written or AI-generated, while the generator produces synthetic text to challenge and improve the discriminator’s accuracy. Classification outputs are evaluated using losses from both models. Metrics such as accuracy and AUC-ROC are monitored during training to track the discriminator's performance. The validation phase focuses on tuning hyperparameters using the validation set to optimize the discriminator. This ensures that the model generalizes well and avoids overfitting.

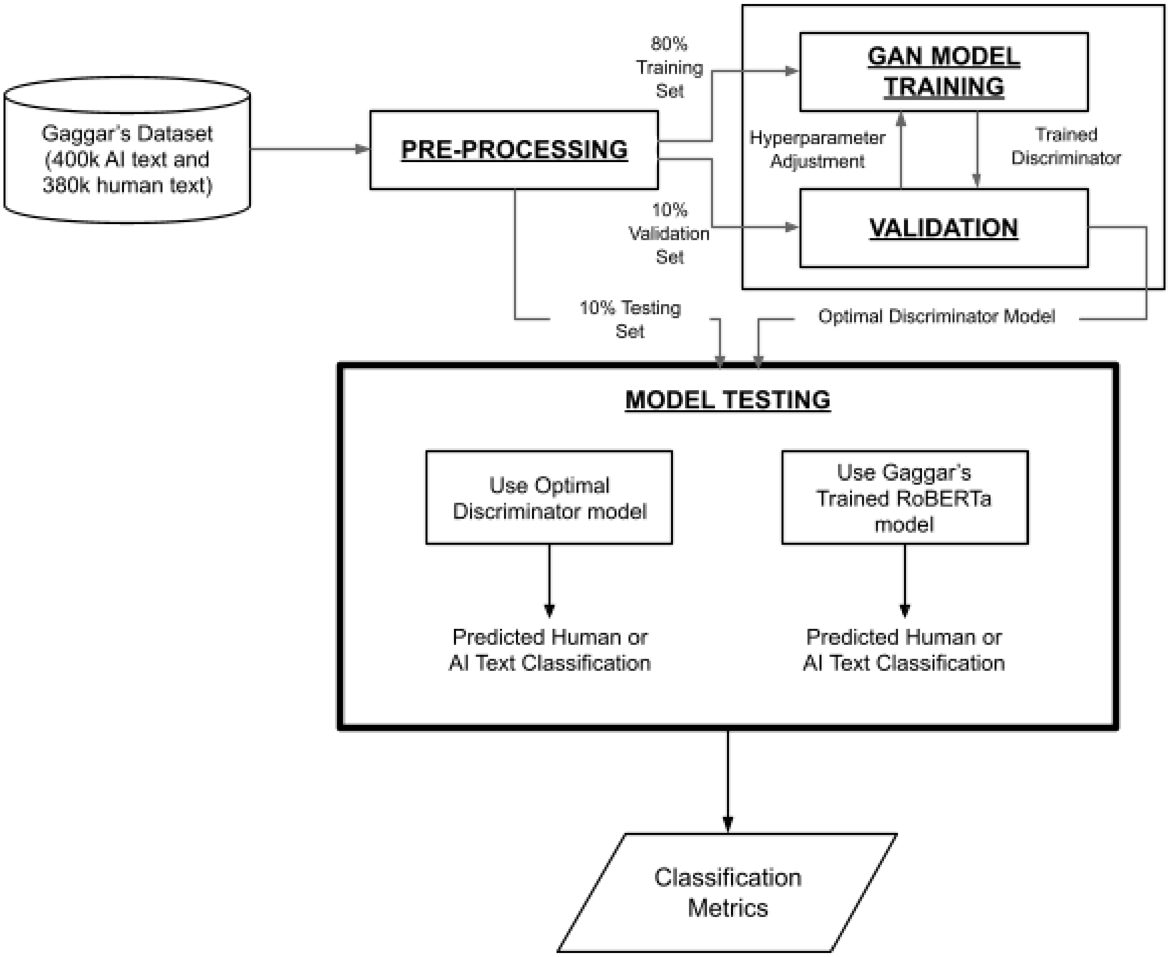

Gaggar's Dataset (400k AI text and 380k human text)
PRE-PROCESSING
80% Training Set
10% Validation Set
10% Testing Set
GAN MODEL TRAINING
Hyperparameter Adjustment
Trained Discriminator
VALIDATION
Optimal Discriminator Model
MODEL TESTING
Use Optimal Discriminator model
Use Gaggar's Trained RoBERTa model
Predicted Human or AI Text Classification
Predicted Human or AI Text Classification
Classification Metrics

Figure 4.1c: System Architecture (Model Testing)

In the model testing phase, the trained discriminator is evaluated using the reserved testing set. The discriminator's ability to classify texts as human or AI is measured through metrics such as accuracy and AUC-ROC. For a comparative analysis, Gaggar et al. 's RoBERTa model is also tested on the same test set. Additional metrics such as precision, recall, and F1-score are computed to provide a comprehensive evaluation of both models. The final outcome is an optimally trained discriminator capable of accurately distinguishing between human-written and AI-generated texts across different domains.

B. Description of the Modules and Interfaces

This section overviews the different modules and interfaces integrated into this study. The input, process, and output details within the pre-processing, training, and testing phases are presented here.

1. Text Cleaning

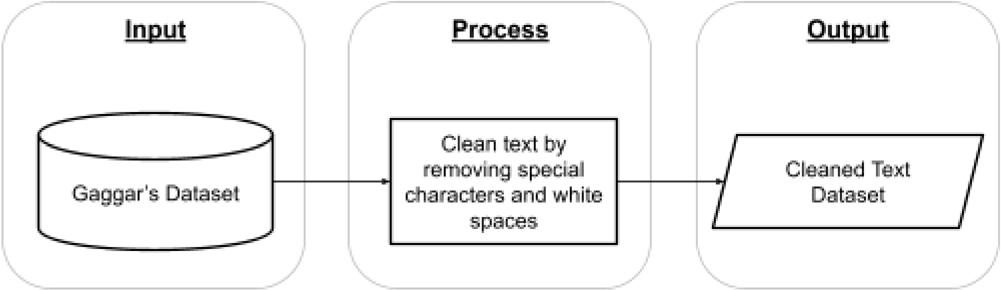


Figure 4.2: Text Cleaning Module

Input: Gaggar et al's Dataset.

Process:

- Removal of special characters and extra spaces - ensuring no empty strings

Output: Cleaned Text Dataset.

2. **Domain Classification**

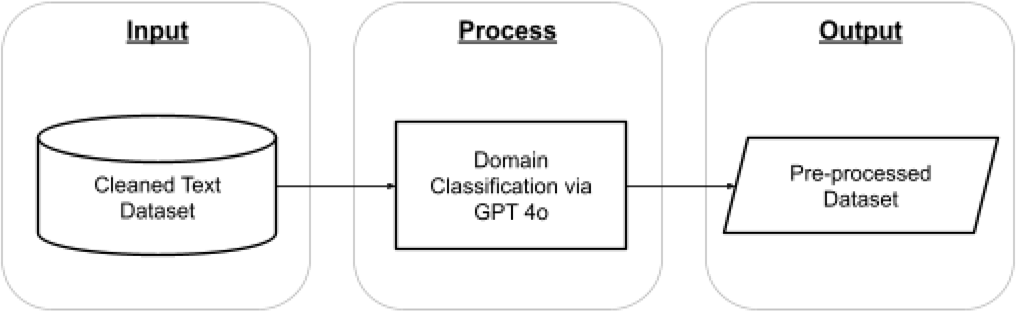


Figure 4.3: Domain Classification Module

Input: Cleaned Text Dataset.

Process:

- Addition of column “Category” for domain classification via GPT 4o.
- GPT 4o classifies each text sample into 8 separate domains/categories.

Output: Pre-processed dataset.

3. **Tokenization**

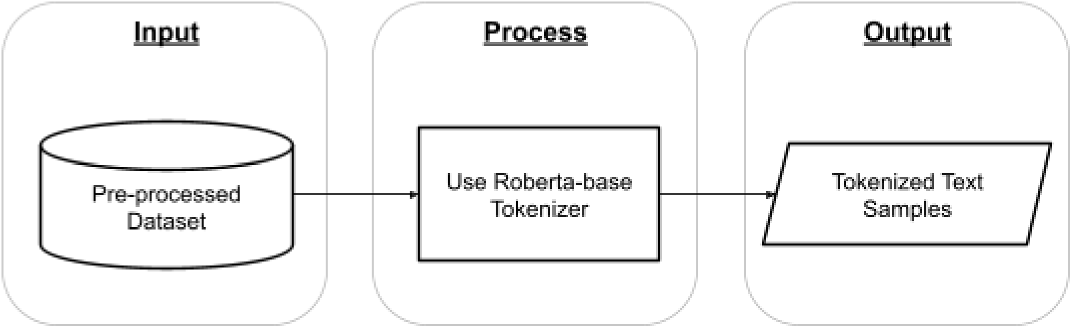


Figure 4.4: Tokenization Module

Input: Pre-processed Dataset.

Process:

- Use the RoBERTa tokenizer to convert text samples into tokens.

- Pad sequences to a uniform length (512 tokens) and create attention masks as part of the tokenization process.

Output: Tokenized Text Samples.

4. **Embedding Generation**

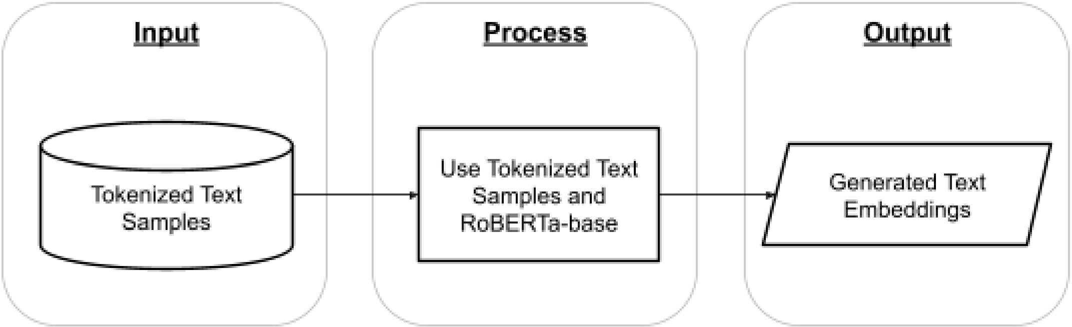

Input
Process
Output
Tokenized Text Samples
Use Tokenized Text Samples and RoBERTa-base
Generated Text Embeddings

Figure 4.5: Embedding Generation Module

Input: Tokenized Text Samples.

Process:

- Use the roberta-base model to convert each tokenized text sample into embeddings by extracting the output from the last hidden state.
- Create an embedding matrix for all vocabulary tokens, which can be used for further GAN processing or analysis.

Output: Generated Text Embeddings.

The purpose of generating text embeddings is to transform the tokenized text samples into a format that the GAN can effectively use for text generation. These embeddings capture the semantic and contextual information of the text, allowing the GAN to generate synthetic samples that are coherent and contextually relevant. Embeddings serve as a crucial input, enabling the generator to produce high-quality

AI-like text that improves the diversity and accuracy of data augmentation for training models.

5. Data Splitting

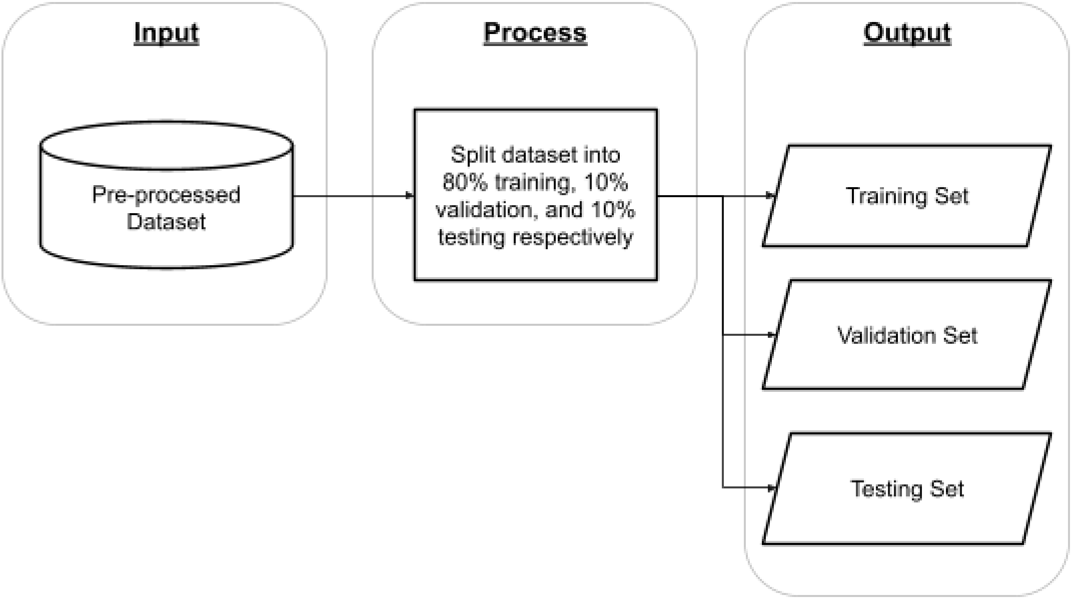

Input
Pre-processed Dataset
Process
Split dataset into 80% training, 10% validation, and 10% testing respectively
Output
Training Set
Validation Set
Testing Set

Figure 4.6: Data Splitting Module

Input: Pre-processed dataset.

Process:

- Shuffle the dataset to ensure randomization.
- Split the dataset into 80% for training, 10% validation and 10% for testing respectively using train_val_test_split to create training, validation, and testing subsets.

Output: 80% Training Set, 10% Validation Set, and 10% Testing Set.

6. GAN Model

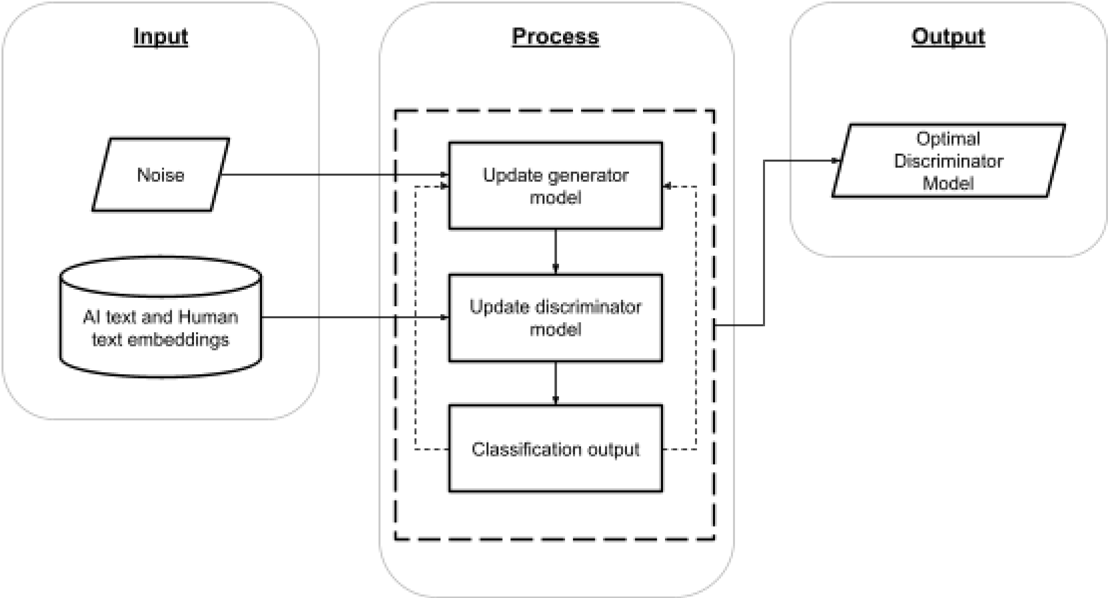

Input
Noise
AI text and Human text embeddings
Process
Update generator model
Update discriminator model
Classification output
Output
Optimal Discriminator Model

Figure 4.7: GAN Model Module

Input: AI text embeddings, Human text embeddings, and Noise.

Process:

- Update the generator model to produce synthetic AI text.
- Update the discriminator model to classify real vs. synthetic text.
- Generate synthetic samples through iterative training.

Output: Optimal Discriminator Model.

For the Generator, noise is used as an input to introduce variability in the synthetic text sentences wherein the generator transforms this noise into a meaningful output.

7. Discriminator Model Training

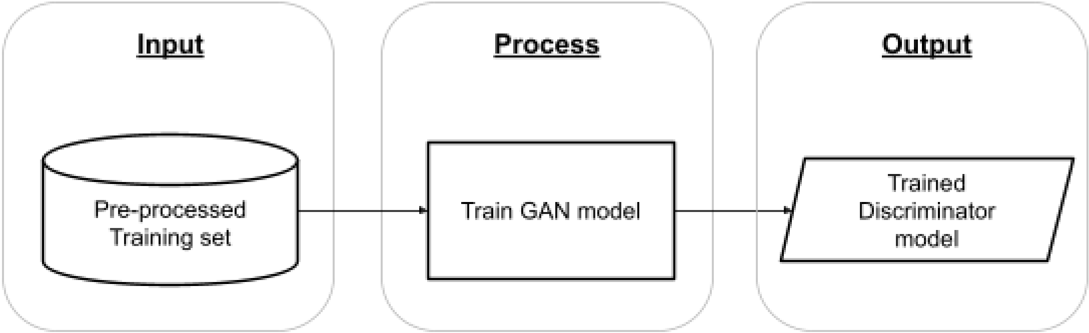


Figure 4.8: Training Module

Input: Pre-processed Training set.

Process:

- Initialize the pretrained GAN model.
- Run the model on the training set for classification or evaluation tasks.

Output: Trained Discriminator model.

8. Discriminator Model Validation

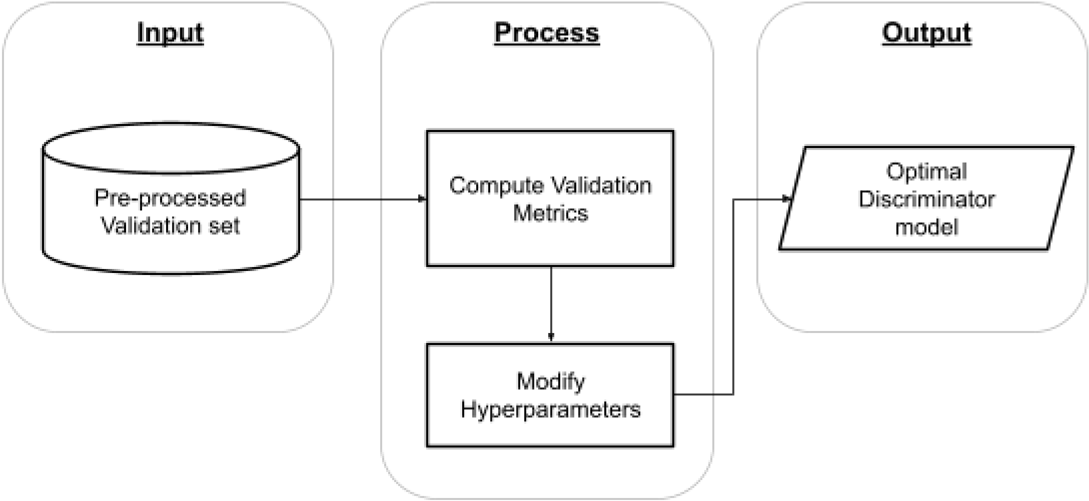

Input
Process
Output
Pre-processed Validation set
Compute Validation Metrics
Modify Hyperparameters
Optimal Discriminator model

Figure 4.9: Validation Module

Input: Pre-processed Validation set.

Process:

- Evaluate the GAN model's performance on the validation set to check generalization.
- Modify hyperparameters to achieve optimal performance.

Output: Optimal Discriminator Model.

9. **Discriminator Model Testing and Evaluation**

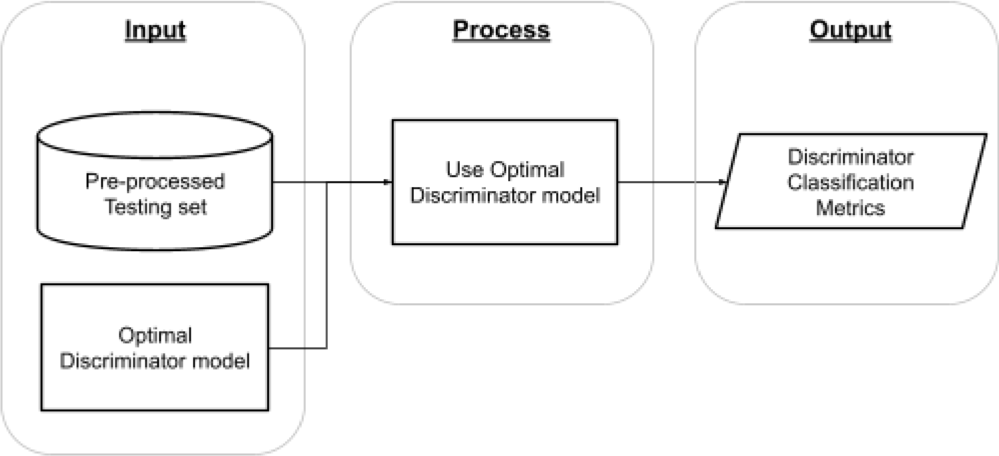

Input
Process
Output
Pre-processed Testing set
Optimal Discriminator model
Use Optimal Discriminator model
Discriminator Classification Metrics

Figure 4.10: Testing Module

Input: Pre-processed Testing set, Optimal Discriminator model.

Process:

- Evaluate the model's performance on the test set to check generalization.
- Calculate classification metrics such as AUC-ROC, Accuracy, Precision, Recall, and F1-score.

Output: Discriminator Model Classification Metrics.

C. Sample System Simulation of Test Data

This section provides sample text examples to demonstrate the system's capability to classify and differentiate between human and AI-generated text. The samples were obtained from Gaggar's Dataset: human-written samples and AI-generated samples, which in this study, text cleaning is applied. The samples illustrate the diversity of language use and structure across these sources.

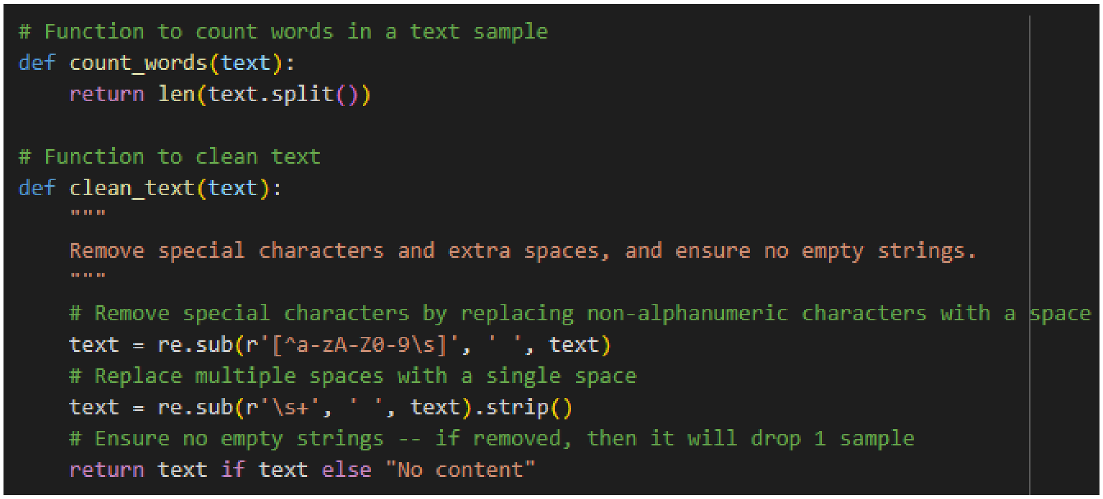
Table 4.1: Test Text Samples of Human and AI text

Source	Human	AI
Gaggar's Dataset	1. Despite the penalty for pass interference on Patrick Pass, the Vikings will need to continue pressuring receivers like Troy Brown and David Pack Rani, while also defending against Edwards' short passes to fullback Marc Edwards, who was stopped just two yards from a first down. Now over to James Brown in Los Angeles. Kimmy, in a recent interview, Steve Spurrier heard Steve Martin complaining about not receiving the ball enough. This game marks Lorenzo's second rushing touchdown. 2. He passes to her left on the 40-yard line, gaining five yards and bringing up 2nd down. The 49ers are	1. This study demonstrates that CaMKIV can relieve STZ induced diabetic neuropathic pain The mechanism of this function depended on the process pCaMKIV localized in the nuclei of DRG neurons and regulated HMGB1 which was an important mediator of neuropathic pain These findings reported CaMKIV may be a potential target or important node in relieving diabetic neuropathic pain 2. These data suggest that alcohol use in patients with bipolar disorder and substance dependence increases the risk of a depressive episode in the near term

	struggling to find their rhythm against the Jets' complex coverage, and Coach Bill Parcells advised them to pressure quarterback Steve Young. On 2nd and 5, Young connects with JJ Stokes for a 24-yard gain to the 32-33 yard line. 3. Consistent with expectations, prevalence of high BMI in this sample of rural Appalachian children exceeds national averages. Prevalence of overweight varied by age and sex; boys are particularly vulnerable to developing obesity, especially as they age. Preliminary survey data suggest that eating breakfast at home and at school and increased hours of television viewing may be associated with higher BMI, especially in younger boys.	3. The mean duration of single PLM might be an appropriate parameter to discriminate between healthy subjects with PLM and patients with RLS High numbers of PLM sequences of short duration might be an indicator for the decreased sleep quality in RLS patients

a. Dataset Text Cleaning

Definition of functions count_words and clean_text. Count_words counts the words in a text. Clean_text function removes non-alphanumeric characters, reduces multiple spaces to one, trims extra spaces, and makes sure there is non-empty output.



```python
# Function to count words in a text sample
def count_words(text):
    return len(text.split())

# Function to clean text
def clean_text(text):
    """
    Remove special characters and extra spaces, and ensure no empty strings.
    """
    # Remove special characters by replacing non-alphanumeric characters with a space
    text = re.sub(r'[^a-zA-Z0-9\s]', ' ', text)
    # Replace multiple spaces with a single space
    text = re.sub(r'\s+', ' ', text).strip()
    # Ensure no empty strings -- if removed, then it will drop 1 sample
    return text if text else "No content"
```

Figure 4.10: Text cleaning code snippet

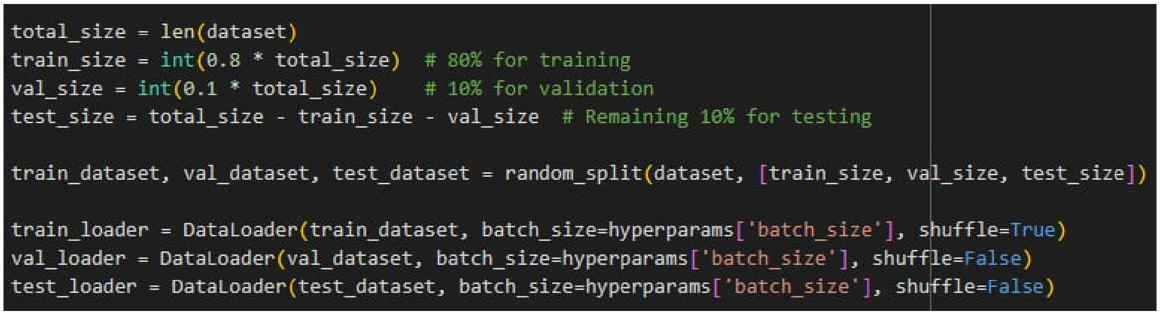
Table 4.2: Cleaned Text Samples

Class	Before Text Cleaning	After Text Cleaning
Human	Why should n't they be cheap ? Shipping long distances by cargo ship is more efficient than most people believe . For example , the current market rate to ship a 40 foot refrigerated container from Brazil to New York is $ 3907 . Such a container could carry approximately 57,579 pounds of bananas . $ 3907/57,579 = 7 cents / pound . SchrÃ¶dinger 's Cat is a thought experiment in applying the idea of	Why should n t they be cheap Shipping long distances by cargo ship is more efficient than most people believe For example the current market rate to ship a 40 foot refrigerated container from Brazil to New York is 3907 Such a container could carry approximately 57 579 pounds of bananas 3907 57 579 7 cents pound Schr dinger s Cat is a thought experiment in applying the idea of

	quantum entanglement to non - quantum objects . There 's not an actual experiment .	quantum entanglement to non quantum objects There s not an actual experiment
AI	" Pros: - Solar energy is renewable and sustainable, meaning that it can be used indefinitely without diminishing. - Solar energy is clean and emits no pollutants or greenhouse gases. - Solar energy is widely available and can be used to generate electricity anywhere in the world on a large or small scale. Cons: - Solar energy typically requires a large upfront investment, as photovoltaic systems are expensive. - Solar energy is intermittent and can be affected by weather conditions, meaning it isnâ€™t always reliable. - Solar energy requires a large amount of area for installation."	Pros Solar energy is renewable and sustainable meaning that it can be used indefinitely without diminishing Solar energy is clean and emits no pollutants or greenhouse gases Solar energy is widely available and can be used to generate electricity anywhere in the world on a large or small scale Cons Solar energy typically requires a large upfront investment as photovoltaic systems are expensive Solar energy is intermittent and can be affected by weather conditions meaning it isn t always reliable Solar energy requires a large amount of area for installation

b. Data Splitting

To identify the optimal data split ratio for training, validation, and testing, an 80:10:10 configuration was applied to the balanced dataset. This split was chosen to ensure sufficient training data, reliable validation for hyperparameter tuning, and an adequate test set for performance evaluation. The impact of this split on model performance metrics, such as accuracy and AUC-ROC, was assessed to ensure a balance between training sufficiency and evaluation reliability.

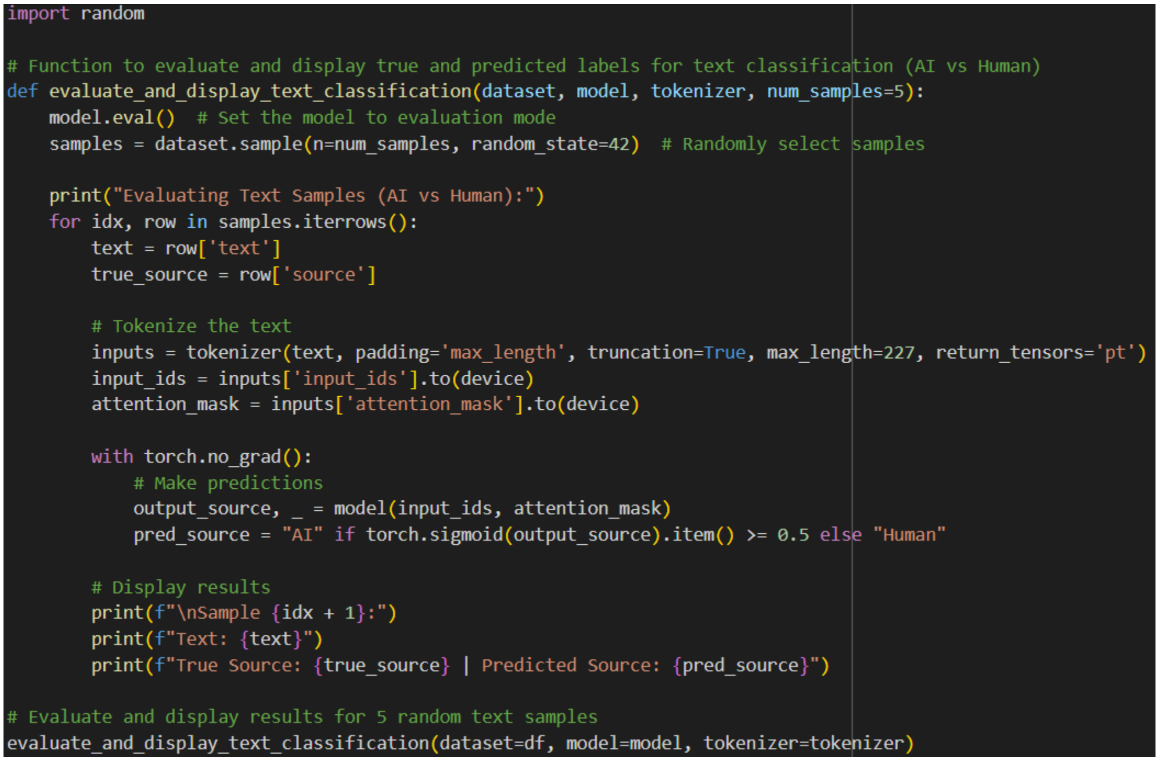


```
total_size = len(dataset)
train_size = int(0.8 * total_size)  # 80% for training
val_size = int(0.1 * total_size)    # 10% for validation
test_size = total_size - train_size - val_size  # Remaining 10% for testing

train_dataset, val_dataset, test_dataset = random_split(dataset, [train_size, val_size, test_size])

train_loader = DataLoader(train_dataset, batch_size=hyperparams['batch_size'], shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=hyperparams['batch_size'], shuffle=False)
test_loader = DataLoader(test_dataset, batch_size=hyperparams['batch_size'], shuffle=False)
```

Figure 4.13: Data Splitting code snippet

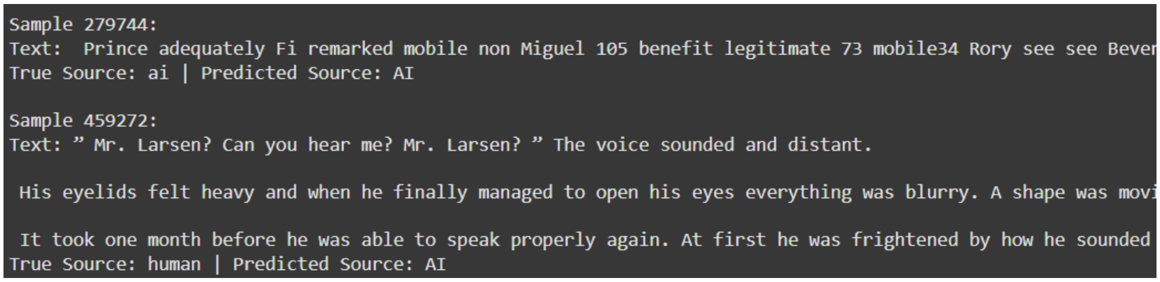
c. Data Processing

This section outlines the process of evaluating and displaying text classification predictions. The presented code snippet demonstrates the implementation of a function that randomly samples text entries from the dataset, processes them through the trained model, and outputs the true and predicted labels. The function employs tokenization and attention mechanisms for input preparation before making predictions using the model.



```
import random

# Function to evaluate and display true and predicted labels for text classification (AI vs Human)
def evaluate_and_display_text_classification(dataset, model, tokenizer, num_samples=5):
    model.eval()  # Set the model to evaluation mode
    samples = dataset.sample(n=num_samples, random_state=42)  # Randomly select samples

    print("Evaluating Text Samples (AI vs Human):")
    for idx, row in samples.iterrows():
        text = row['text']
        true_source = row['source']

        # Tokenize the text
        inputs = tokenizer(text, padding='max_length', truncation=True, max_length=227, return_tensors='pt')
        input_ids = inputs['input_ids'].to(device)
        attention_mask = inputs['attention_mask'].to(device)

        with torch.no_grad():
            # Make predictions
            output_source, _ = model(input_ids, attention_mask)
            pred_source = "AI" if torch.sigmoid(output_source).item() >= 0.5 else "Human"

        # Display results
        print(f"\nSample {idx + 1}:")
        print(f"Text: {text}")
        print(f"True Source: {true_source} | Predicted Source: {pred_source}")

# Evaluate and display results for 5 random text samples
evaluate_and_display_text_classification(dataset=df, model=model, tokenizer=tokenizer)
```

Figure 4.16: Text Classification code snippet



```
Sample 279744:
Text:  Prince adequately Fi remarked mobile non Miguel 105 benefit legitimate 73 mobile34 Rory see see Bever
True Source: ai | Predicted Source: AI

Sample 459272:
Text: ” Mr. Larsen? Can you hear me? Mr. Larsen? ” The voice sounded and distant.

 His eyelids felt heavy and when he finally managed to open his eyes everything was blurry. A shape was movi

 It took one month before he was able to speak properly again. At first he was frightened by how he sounded
True Source: human | Predicted Source: AI
```

Figure 4.17: Classified pre-processed text sample

D. Test Results

This section presents the performance of the GAN Discriminator on the testing dataset for the binary classification task of distinguishing between AI-generated and human-written text. Evaluation metrics such as accuracy, precision, recall, F1-score, and AUC-ROC are reported to assess the model's effectiveness in identifying the text source. Additionally, a confusion matrix is included to visualize the model's predictions and errors. The results highlight the discriminator's performance and the effectiveness of the training process in achieving accurate text classification.

a. GAN Training Phase

i. Learning Rates of GAN model

This subsection examines how varying learning rates affect the training performance of GAN-RoBERTa. Adjusting the generator (g_lr) and discriminator (d_lr) learning rates helps balance fast learning with training stability. These adjustments optimize the generator's ability to produce realistic synthetic text and the discriminator's classification accuracy. Table 4.7 provides the discriminator and generator losses for different learning rate combinations.

Table 4.7: Losses of different Learning Rates

Learning Rate	Discriminator Loss	Generator Loss
g_lr: 0.000000009 d_lr: 0.000000004	1.3767	2.2575
g_lr: 0.00000009 **d_lr: 0.00000004**	1.3406	2.5208
g_lr: 0.0000009 d_lr: 0.0000004	1.2972	2.8428

Table 4.7 presents the discriminator and generator losses during GAN training for different combinations of learning rates. As the learning rate increases from *g_lr: 0.000000009* and *d_lr: 0.000000004* to *g_lr: 0.0000009* and d_lr: 0.0000004 he discriminator loss gradually decreases (from 1.3767 to 1.2972), indicating an improved ability to distinguish real from synthetic text. On the other hand, the generator loss increases (from 2.2575 to 2.8428), reflecting greater difficulty for the generator in adapting to the discriminator improving performance.

These results suggest that higher learning rates stabilize the discriminator's effectiveness, while the generator continues to improve its synthetic text generation.

Table 4.8: Summary of different Learning Rates

Learning Rate	Self-BLEU Score	NLL Score
g_lr: 0.000000009 d_lr: 0.000000004	0.6987	9.9635
g_lr: 0.00000009 **d_lr: 0.00000004**	0.6645	9.9634
g_lr: 0.0000009 d_lr: 0.0000004	0.6877	9.9634

Table 4.8 provides an evaluation of GAN training under varying learning rates, focusing on the quality of generated samples. The table includes the Self-BLEU Score, which measures diversity among generated samples, and the NLL (Negative Log-Likelihood) Score, which assesses how well the generator captures the real data distribution. As the learning rate increases, there is a noticeable fluctuation in the Self-BLEU scores (ranging from 0.6645 to 0.6987) while the NLL scores remain stable at 9.9634.

The generated samples show a mix of coherent and nonsensical phrases. Lower learning rates appear to produce more structured outputs, whereas higher learning rates give more randomness, indicating a trade-off between diversity and consistency.

ii. Dropout Rates of GAN model

This subsection explores how varying dropout rates affect the training performance of GAN-RoBERTa. Dropout is a regularization technique that prevents overfitting by randomly deactivating a portion of neurons during training. Adjusting the dropout rates for the generator (g_dropout) and discriminator (d_dropout) ensures an optimal trade-off between model complexity and generalization. The study evaluates different dropout rate combinations to assess their effect on discriminator and generator losses. Table 4.9 provides a summary of the discriminator and generator losses for different dropout rate settings.

Table 4.9: Losses of different Dropout Rates

Dropout Rates	Discriminator Loss	Generator Loss
g_dropout: 0.42 d_dropout: 0.40	1.3404	2.5178
g_dropout: 0.46 **d_dropout: 0.44**	1.3406	2.5208
g_dropout: 0.50 d_dropout: 0.48	1.3409	2.5205

Table 4.9 presents the discriminator and generator losses during GAN training for various dropout rate settings. As the dropout rates increase from *g_dropout: 0.42* and *d_dropout: 0.40* to *g_dropout: 0.50* and *d_dropout: 0.48,* the discriminator loss remains relatively stable, ranging from 1.3404 to 1.3409. Similarly, the generator loss shows

minimal fluctuation, staying between 2.5178 and 2.5208. These results indicate that varying the dropout rates within this range does not significantly impact the losses, suggesting that the model maintains consistent training performance across these configurations.

Table 4.10: Summary of different Dropout Rates

Dropout Rates	Self-BLEU Score	NLL Score
g_dropout: 0.42 d_dropout: 0.40	0.6852	10.4312
g_dropout: 0.46 **d_dropout: 0.44**	0.6645	9.9634
g_dropout: 0.50 d_dropout: 0.48	0.6766	9.9635

Table 4.10 evaluates the impact of varying dropout rates on the quality and diversity of generated text in GAN training for GAN-RoBERTa. As the dropout rate increases, the Self-BLEU score fluctuates between 0.6645 and 0.6852, indicating varying levels of diversity in the generated text. Meanwhile, the NLL scores remain relatively stable, ranging from 9.9634 to 10.4312, suggesting consistent alignment with the target data distribution.

The generated text samples show differences in consistency and randomness across dropout rate settings. Lower dropout rates produce slightly more structured text, while higher rates increase randomness.

iii. Batch Sizes of GAN model

This subsection shows the impact of batch sizes on the training performance of GAN-RoBERTa. Smaller batch sizes typically provide more frequent updates but may lead to unstable training, while larger batch sizes offer smoother gradients but require more computational resources. This section evaluates different batch sizes to understand their influence on the discriminator and generator losses of the GAN-RoBERTa model.

Table 4.11: Losses of different Batch Sizes

Batch Sizes	**Discriminator Loss**	**Generator Loss**
batch_size: 32	1.3283	2.6193
batch_size: 64	1.3406	2.5208
batch_size: 128	1.3630	2.3557

Table 4.11 presents the discriminator and generator losses during GAN training for varying batch sizes. As the batch size increases from 32 to 128, the discriminator loss shows a gradual increase (from 1.3283 to 1.3630), indicating a slightly reduced ability to distinguish between real and synthetic samples. Conversely, the generator loss decreases (from 2.6193 to 2.3557), suggesting improved performance in generating synthetic text as larger batch sizes allow for more stable gradient updates.

These results highlight the trade-offs between smaller and larger batch sizes. Smaller batch sizes provide more frequent updates, potentially leading to finer discriminator performance, while larger batch sizes result in smoother training dynamics that benefit the generator's performance.

Table 4.12: Summary of different Batch Sizes

Batch Sizes	**Self-BLEU Score**	**NLL Score**
batch_size: 32	0.6928	9.6540
batch_size: 64	0.6645	9.9634
batch_size: 128	0.6693	10.0035

Table 4.12 shows how batch size influences the quality and variability of text generated by the GAN model in GAN-RoBERTa. The Self-BLEU score decreases slightly as the batch size increases from 32 to 128, reflecting minor improvements in sample diversity. Conversely, the NLL score shows a gradual increase, indicating a subtle decline in the generator's ability to match the true data distribution at larger batch sizes.

The generated text samples reveal how batch sizes affect the structure and randomness of generated text. Smaller batch sizes produce outputs with greater variation, while larger batch sizes favor structured but less diverse samples.

iv. Warm Up Epochs of GAN model

This subsection examines the effect of warm-up epochs on the training performance of GAN-RoBERTa. Warm-up epochs gradually increase the learning rate during the initial phase of training to stabilize the model's convergence and prevent abrupt changes in weights. This approach is particularly useful for GANs, where balanced training between the generator and discriminator is essential. Different warm-up epoch settings are evaluated to understand their impact on discriminator and generator losses.

Table 4.13: Losses of different Warm Up Epochs

Warm Up Epochs	**Discriminator Loss**	**Generator Loss**
warmup_epochs: 30	1.3404	2.5081
warmup_epochs: 50	1.3406	2.5208
warmup_epochs: 70	1.3446	2.5065

Table 4.13 presents the discriminator and generator losses across varying numbers of warm-up epochs during GAN-RoBERTa training. As the warm-up epochs increase from 30 to 70, the discriminator loss shows a slight rise (from 1.3404 to 1.3446), indicating minimal changes in its ability to differentiate real from synthetic samples. Meanwhile, the generator loss fluctuates slightly, with values ranging between 2.5065 and 2.5208, suggesting consistent performance in generating synthetic text.

Table 4.14: Summary of different Warm Up Epochs

Warm Up Epochs	**Self-BLEU Score**	**NLL Score**
warmup_epochs: 30	0.7832	9.9633
warmup_epochs: 50	0.6645	9.9634
warmup_epochs: 70	0.7594	9.5189

Table 4.14 evaluates the impact of varying warm-up epochs on the diversity and quality of generated text in GAN-RoBERTa training. The Self-BLEU score ranges from 0.6645 to 0.7832, reflecting slight changes in text diversity as the number of warm-up epochs increases. The NLL score shows minimal fluctuation, staying between 9.5189 and 9.9634, shows that the generator consistently approximates the target data distribution.

E. Analysis and Interpretation of the Results

The analysis focuses on evaluating the performance of Gaggar's RoBERTa in text classification tasks versus the use of the trained GAN Discriminator on the same dataset, including its detection performance in the texts of different domains. This section provides insights into statistical comparisons, highlighting key differences in testing accuracy and AUC-ROC metrics.

a. Statistical Analysis

i. Text Classification Train-Val-Test Results

Table 4.15: AUC-ROC Results for RoBERTa vs. GAN

Model (0.5 threshold)	**Train**	**Validation**	**Test**
RoBERTa-base (Gaggar)	96.85%	**95.53%**	95.24%
RoBERTa-base (Replicated model)	**97.62%**	95.24%	**97.31%**
GAN-Discriminator	93.03%	92.71%	92.55%

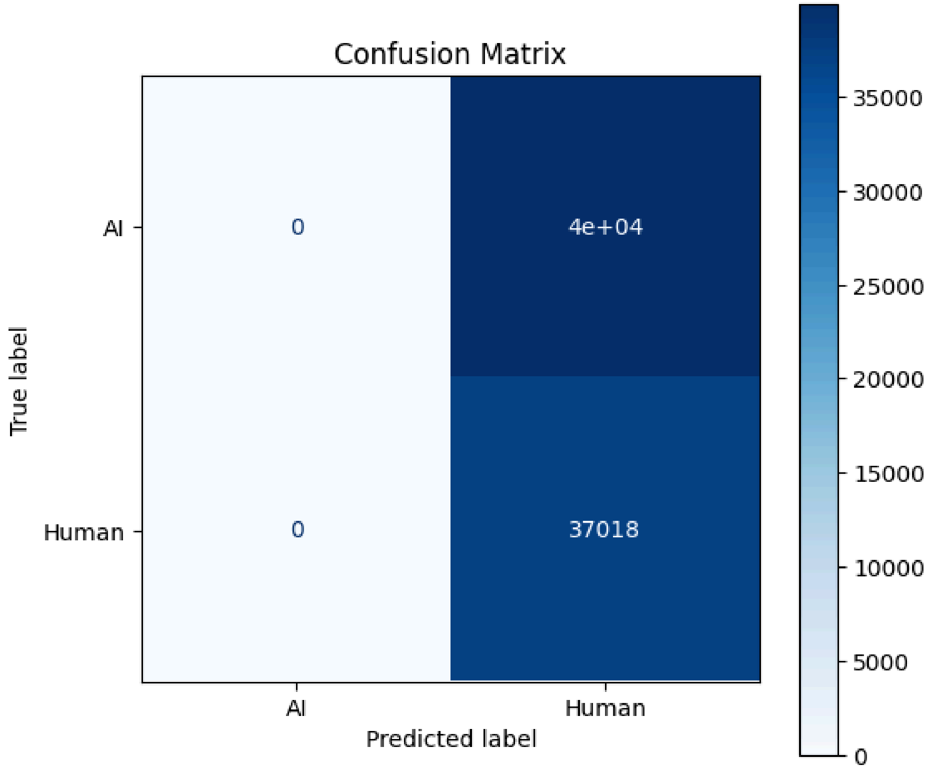
Table 4.15 presents the AUC-ROC results for text classification using the RoBERTa-base model from Gaggar's study, RoBERTa-base replicated model and the GAN-Discriminator. The RoBERTa-base model, as reported in the original study, achieved AUC-ROC scores of 96.85%, 95.24%, and 95.53% on the train, validation, and test sets, respectively. In contrast, when retrained, the replicated RoBERTa-base model achieved improved scores of 97.62%, 95.24%, and 99.31% across the same datasets, demonstrating better performance on the test set. Meanwhile, the GAN-Discriminator achieved 98.13%, 98.05%, and 97.98% for the train, validation, and test sets, respectively.

ii. Thresholding

Table 4.16: Results in different Threshold values for GAN-Discriminator

0.5		**0.6**		**0.7**		**0.8**	
Accuracy	**AUC-ROC**	**Accuracy**	**AUC-ROC**	**Accuracy**	**AUC-ROC**	**Accuracy**	**AUC-ROC**
48.09%	92.55	**81.59%**	**92.71**	51.87%	92.82	51.87%	92.73

Thresholding significantly influences the performance of the GAN-Discriminator, as shown in Table 4.16. At the default threshold of 0.5, the model achieves a high AUC-ROC of 92.% but a low accuracy of 48.09%, likely due to a bias toward misclassifying AI-generated samples as human-written. Increasing the threshold to 0.6 significantly improves the accuracy to 81.59%, while the AUC-ROC remains consistent at at the 92% mark, around 92.71%, indicating a better balance between precision and recall. Raising the threshold further to 0.7 and 0.8 results in the model predictions becoming overly conservative – labeling every input as AI and boosting AUC-ROC slightly (to about 92.8%) but caused the overall accuracy to drop down to 51.87%. In other words, while pushing the threshold values higher reduced false positive cases of the AI class, it also completely eliminated the model's ability to correctly identify human texts.



Figures 4.18: GAN Confusion Matrix for 0.5 Threshold

Figure 4.18 illustrates the confusion matrix for the GAN-Discriminator at a 0.5 threshold. The model completely misclassified all AI-generated samples as human-written, resulting in zero true positives for the AI class. However, it correctly identifies all human-written samples, showing a strong bias toward the human class. This explains the low accuracy despite the high AUC-ROC score, highlighting the need for optimized thresholding to address class imbalance and improve overall performance.

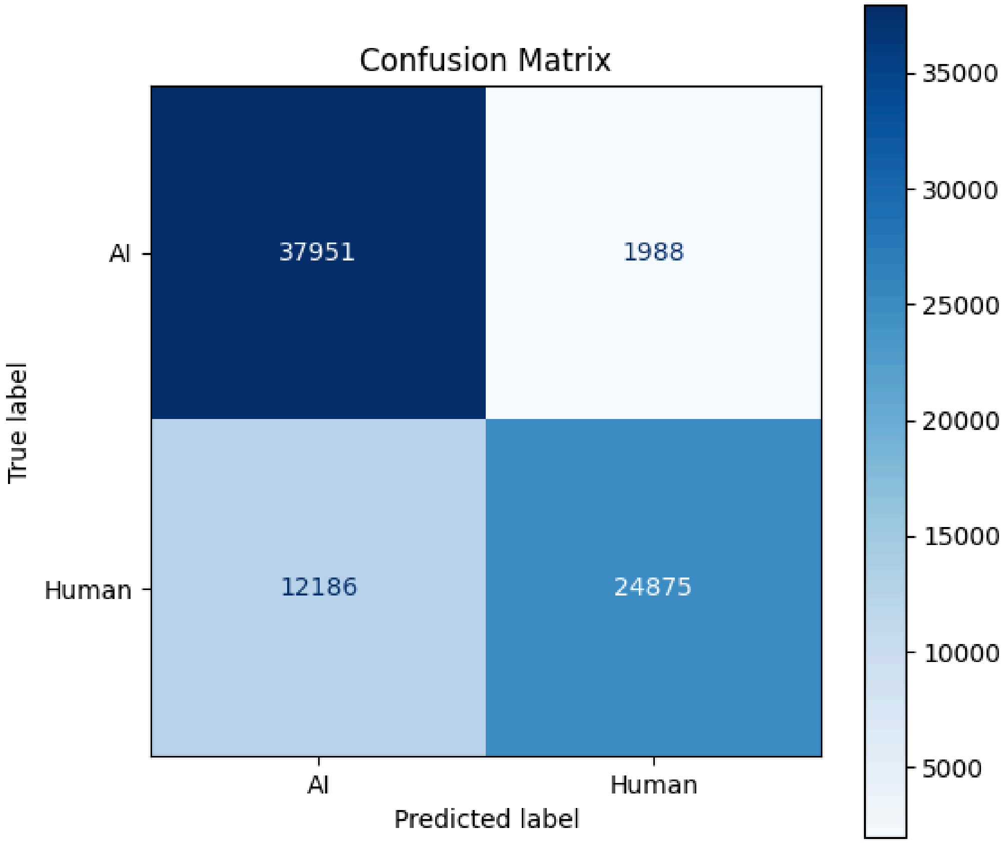


Figure 4.19: GAN Confusion Matrix for 0.6 Threshold

Figure 4.19 presents the confusion matrix for the GAN-Discriminator at a 0.6 threshold. The model demonstrates strong performance in classifying AI-generated samples, correctly predicting 37,951 AI instances with only 1,988 misclassified as human. However, 12,186 human-written samples are misclassified as AI, while 24,875 are correctly identified. This indicates that the model achieves higher sensitivity for AI classification but struggles slightly with human text, leading to a greater number of false positives. Despite this, the overall accuracy remains robust, reflecting the model's ability to distinguish between AI-generated and human-written text effectively. These results emphasize the need for careful threshold tuning to optimize the balance between sensitivity and specificity.

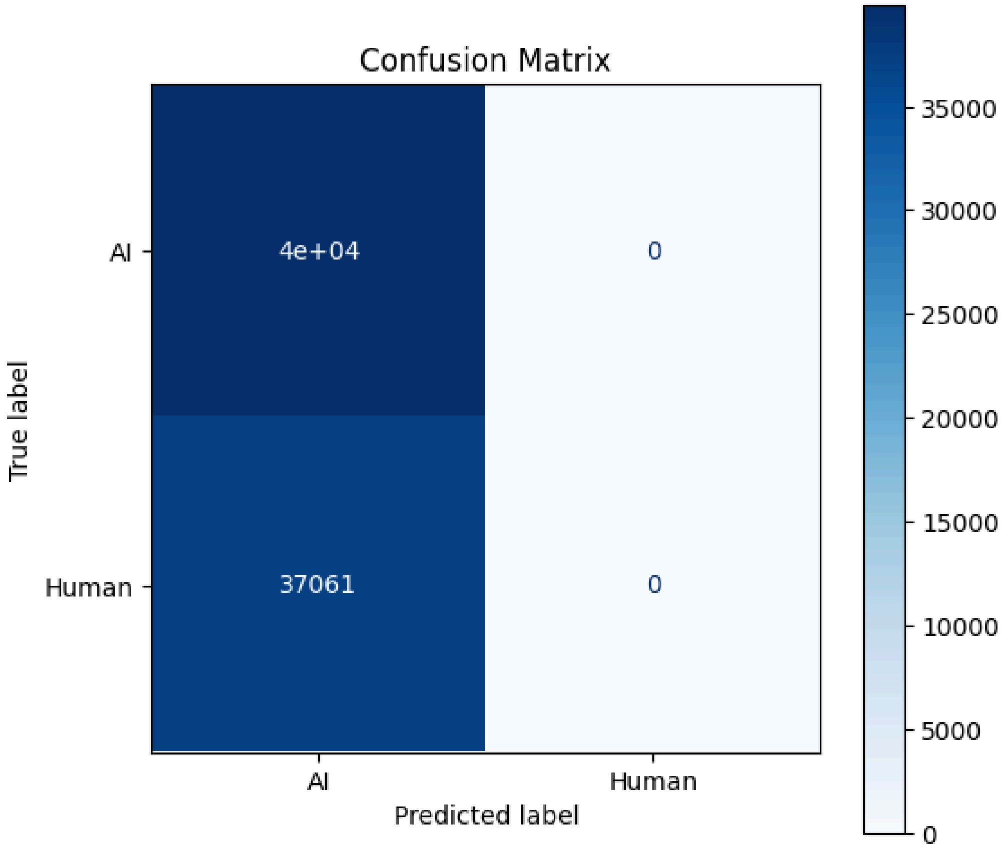


Figure 4.20: GAN Confusion Matrix for 0.7 Threshold

Figure 4.20 presents the confusion matrix for the GAN-Discriminator at a 0.7 threshold. At this threshold, the model correctly classifies all 40,000 AI-generated samples as AI but misclassifies all 37,061 human-written samples as AI. This results in a complete bias toward the AI class, with zero true positives for human-written text. While the model achieves perfect recall for the AI class, it fails to balance performance across both classes, highlighting the limitations of increasing the threshold too high.

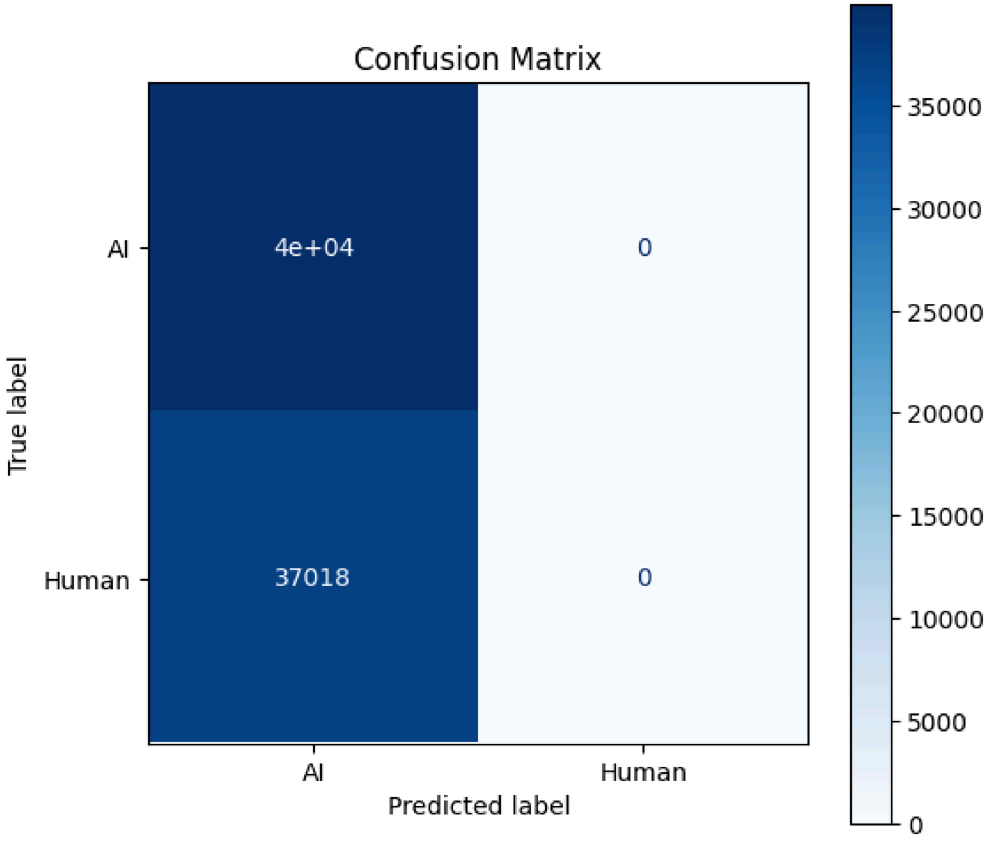


Figure 4.21: GAN Confusion Matrix for 0.8 Threshold

Figure 4.21 shows the confusion matrix for the GAN-Discriminator at a 0.8 threshold. At this threshold, the model classifies all 40,000 AI-generated samples as AI but fails to identify any human-written samples, misclassifying all 37,061 human-written samples as AI. This indicates an extreme bias toward predicting the AI class, similar to the behavior observed at the 0.7 threshold, with no correct predictions for the human class. While the model achieves perfect recall for AI samples, it completely sacrifices performance on human-written text, further emphasizing the limitations of higher thresholds in maintaining class balance.

iii. Text Classification Test Evaluation Metrics

Table 4.16: Text Classification Evaluation Metrics

Metric (0.6 threshold)	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Human	**81.59%**	0.926	0.671	0.778
AI		0.757	0.950	0.843

Table 4.16 presents the evaluation metrics at a 0.6 threshold, showing that for human-written text, the classifier achieves high precision (0.926) but only moderate recall (0.671), resulting in an F1-score of 0.778. In contrast, for AI-generated text, it attains lower precision (0.757) but significantly higher recall (0.950), yielding an F1-score of 0.843. This imbalance is reflected in the accuracy rates: while the model is highly accurate on AI content (95.02%), its human-text accuracy is lower (67.72%), suggesting that at this threshold, it tends to be more conservative in labeling text as human, inadvertently missing a portion of genuine human samples. Overall, these results indicate strong AI-detection capabilities but highlight the need for further refinement to improve the classifier's performance on human-written text.

iv. Text Classification per Domain

To evaluate the model's ability to generalize its classification performance across various contexts, the dataset was divided into eight distinct categories representing different domains. Each domain contains a varying number of samples to reflect its unique characteristics and linguistic patterns. This analysis aims to assess the GAN-Discriminator's performance in detecting AI-generated text within specific domains and provides insights into the model's strengths and limitations across diverse contexts. The distribution of samples across these domains is presented in the subsequent table, offering a detailed view of dataset representation and balance.

Table 4.17: Number of text samples per domain

Domain/Category	Number of Text Samples
Arts/Culture	231,891
Business/Economics	176,881
Education	35,182
Health	26,252
Lifestyle	19,965
Politics	13,640
Science/Technology	13,593
Sports	11,483

1. Arts/Culture

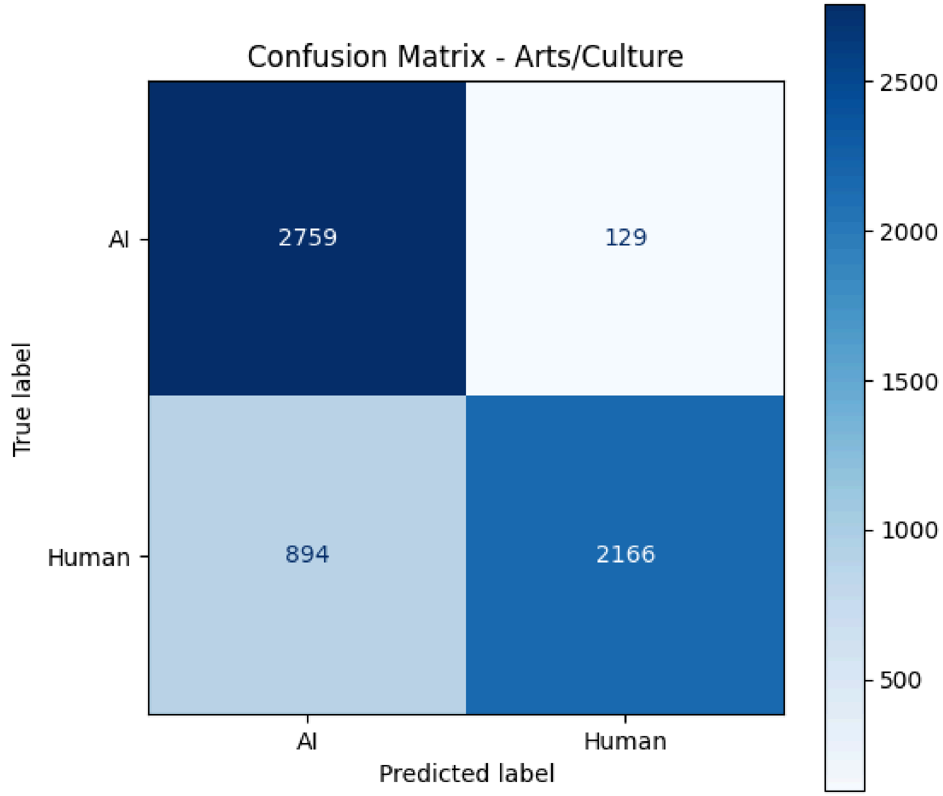


Figure 4.24: Confusion Matrix for Arts/Culture with Text Classification

The confusion matrix for this domain shows that the model performs relatively well with 2759 AI texts and 2166 human texts correctly classified. However, there were 129 AI texts misclassified as Human and 894 human texts as AI suggesting that these may have come from stylistic overlaps in the creative nature of the Arts/Culture domain, where the linguistic features of AI and human-generated texts are similar to each other.

2. Business/Economics

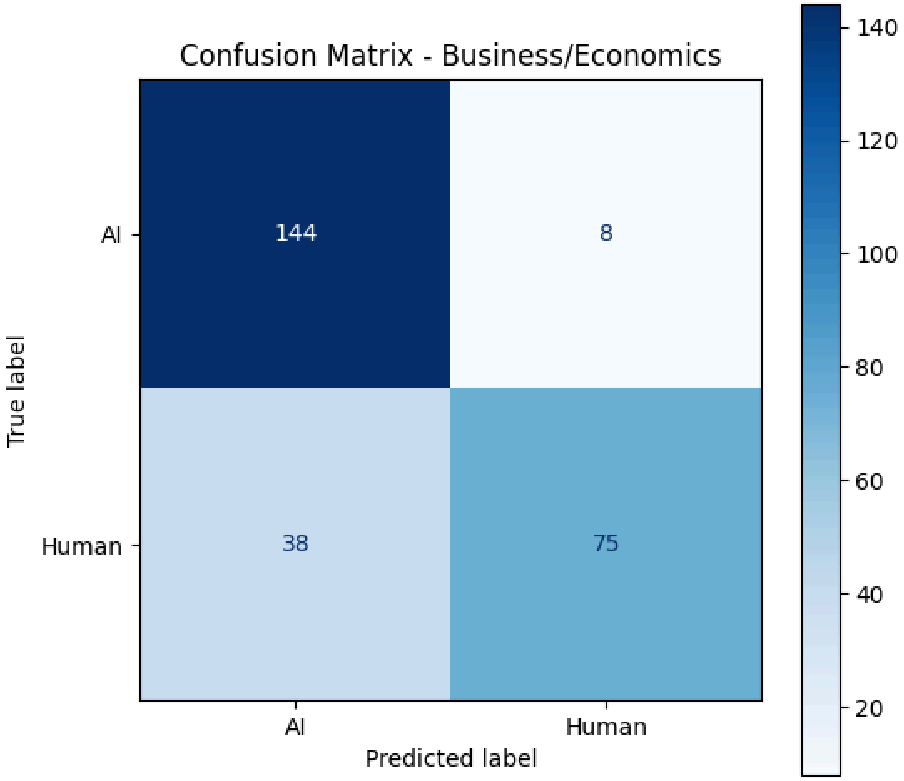


Figure 4.25: Confusion Matrix for Business/Economics with Text Classification

Strong classification performance was observed in the Business/Economics domain with 144 AI texts and 75 human texts correctly identified. There are 8 misclassifications of AI text as human and 35 for human texts as AI. There is an indication of the model being effective in distinguishing between the two classes in this domain which is likely due to the more structured and formal nature of Business/Economic texts.

3. Education

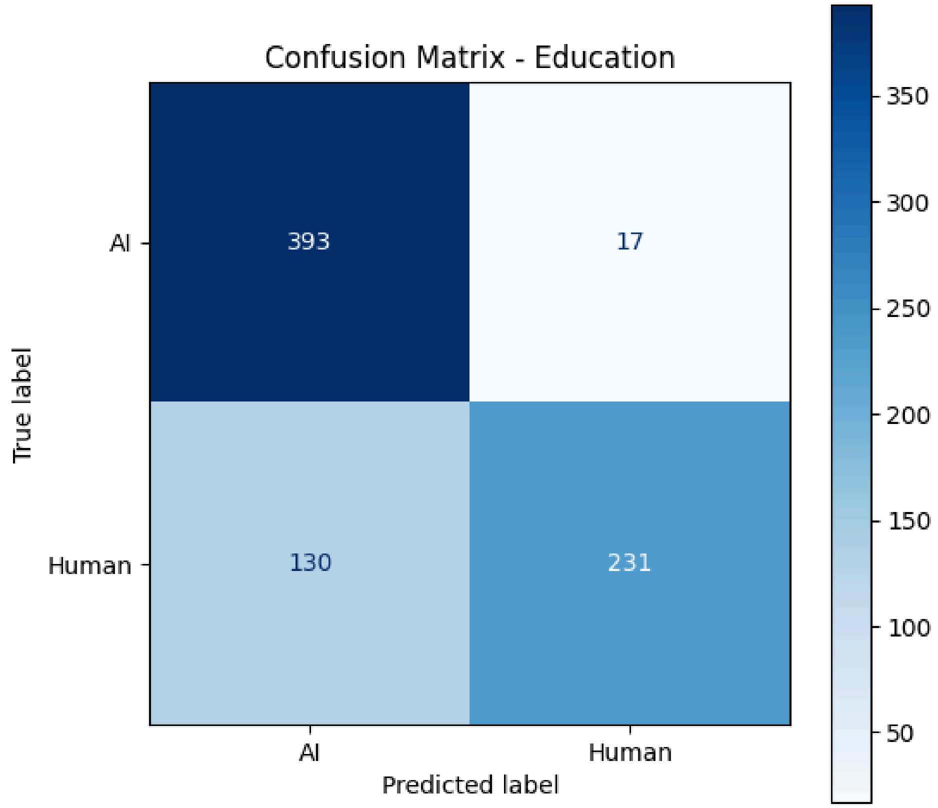


Figure 4.26: Confusion Matrix for Education with Text Classification

The confusion matrix for the Education category highlights strong classification performance, with 393 AI texts and 231 human textscorrectly classified. However, there were 17 AI texts misclassified as human and 130 human texts misclassified as AI. These misclassifications may be attributed to overlapping linguistic features such as formal structure and terminology, which are often common in educational content. Despite these minor errors, the overall accuracy remains high, demonstrating the model's ability to distinguish between AI-generated and human-written educational texts effectively.

4. Health

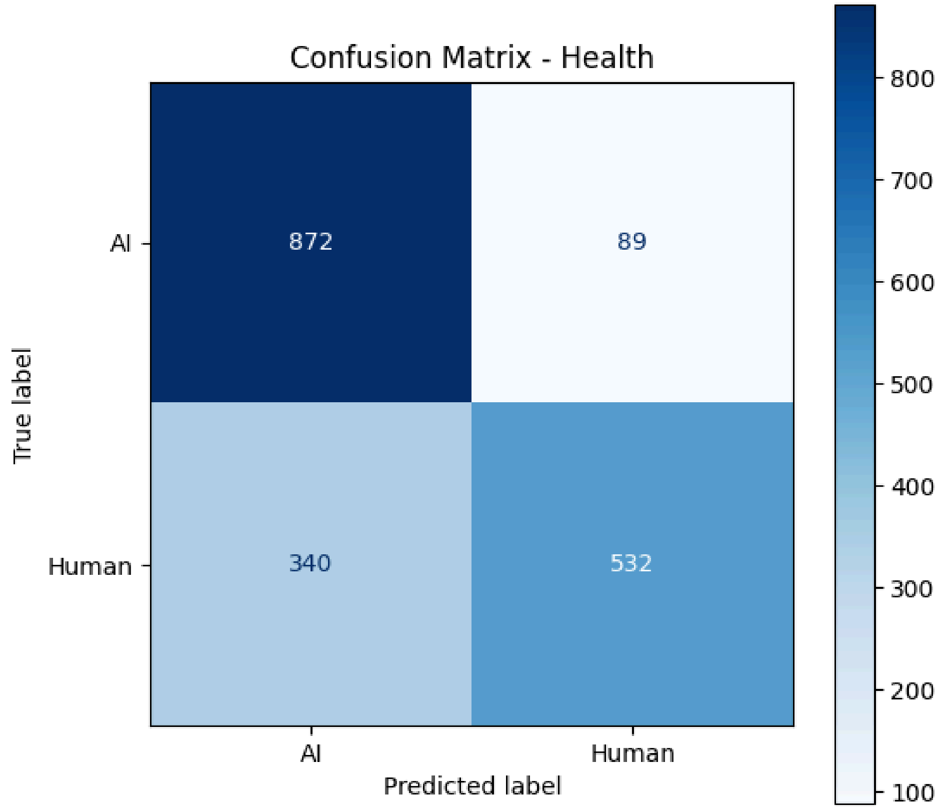


Figure 4.27: Confusion Matrix for Health with Text Classification

The confusion matrix for the Health category shows strong classification performance, with 872 AI texts and 532 human texts correctly classified. However, there were 89 AI texts misclassified as human and 340 human texts misclassified as AI. These misclassifications suggest that certain linguistic patterns or terminology in health-related content may overlap between AI-generated and human-written texts, leading to challenges in distinguishing them accurately. Nonetheless, the overall performance demonstrates reliable classification capabilities for this category.

5. Lifestyle

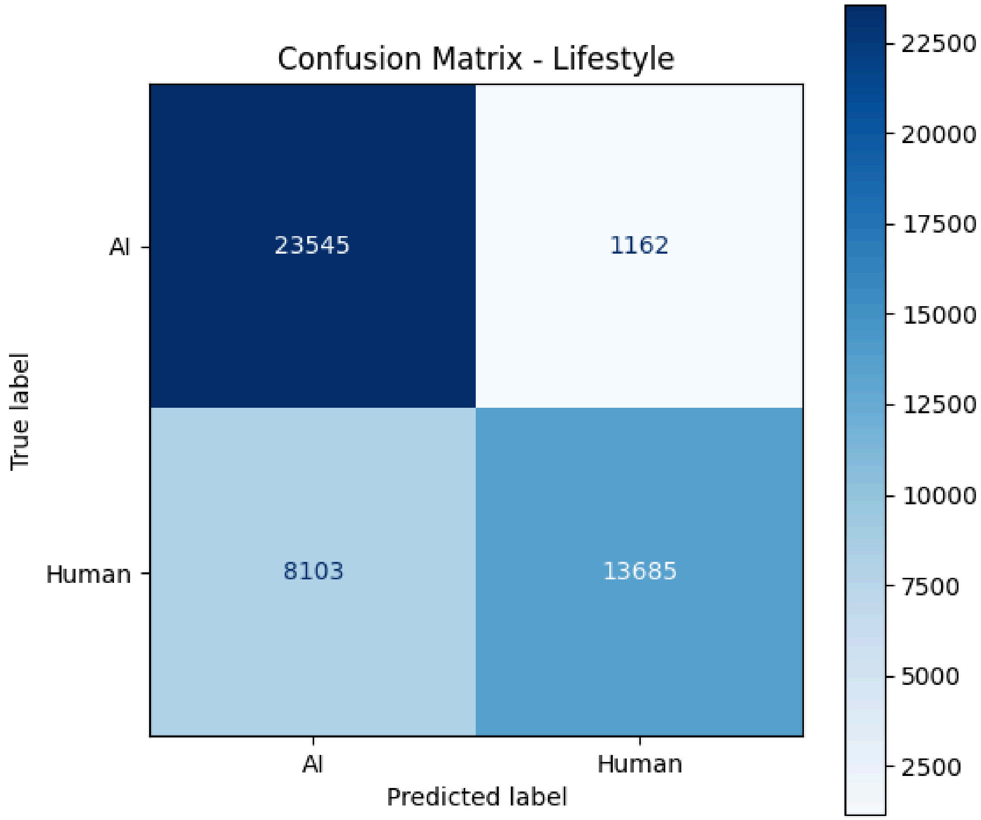


Figure 4.28: Confusion Matrix for Lifestyle with Text Classification

The confusion matrix for the Lifestyle category demonstrates strong classification performance, with 23,545 AI texts and 13,685 human texts correctly classified. However, there were 1,162 AI texts misclassified as human and 8,103 human texts misclassified as AI. The higher misclassification of human texts suggests that AI-generated texts in the Lifestyle category may share stylistic or thematic similarities with human-written content, contributing to the model's difficulty in distinguishing them. Despite this, the majority of predictions are accurate, reflecting reliable overall performance.

6. Politics

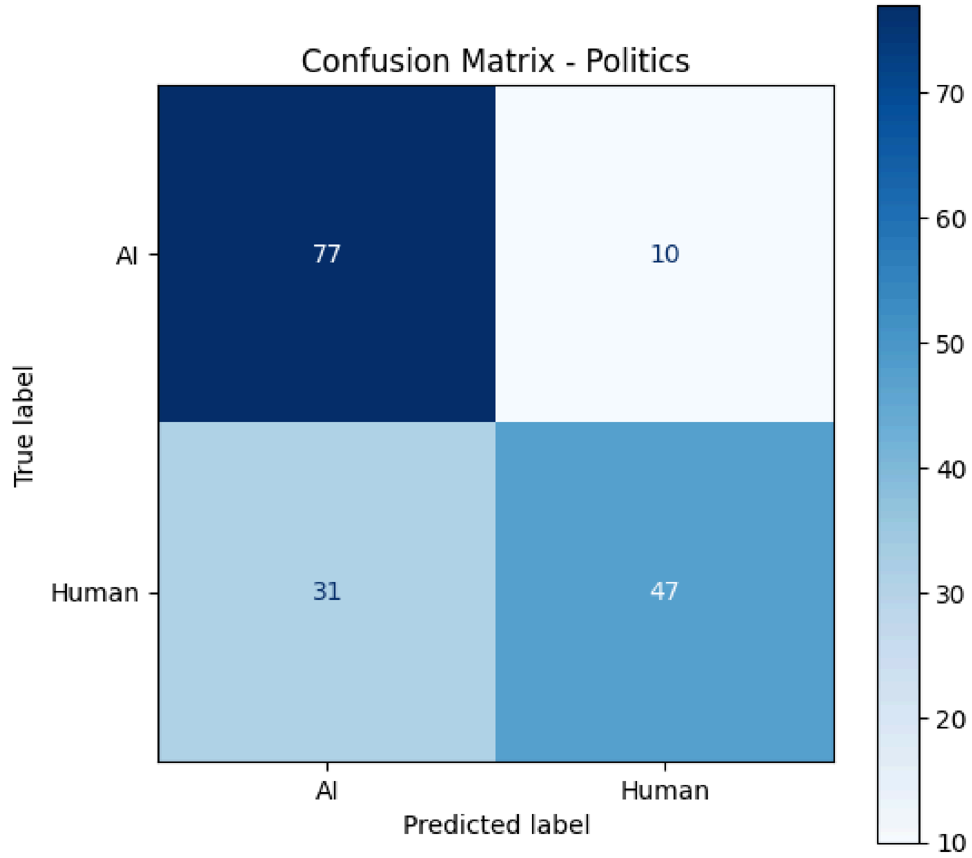


Figure 4.29: Confusion Matrix for Politics with Text Classification

The confusion matrix for the Politics category shows that 77 AI texts and 47 human texts were correctly classified. However, 10 AI texts were misclassified as human, and 31 human texts were misclassified as AI. These misclassifications suggest some overlap in linguistic patterns or terminology between AI-generated and human-written political texts, which could account for the model's difficulty in distinguishing them. Despite this, the model performs reasonably well overall.

7. Science/Technology

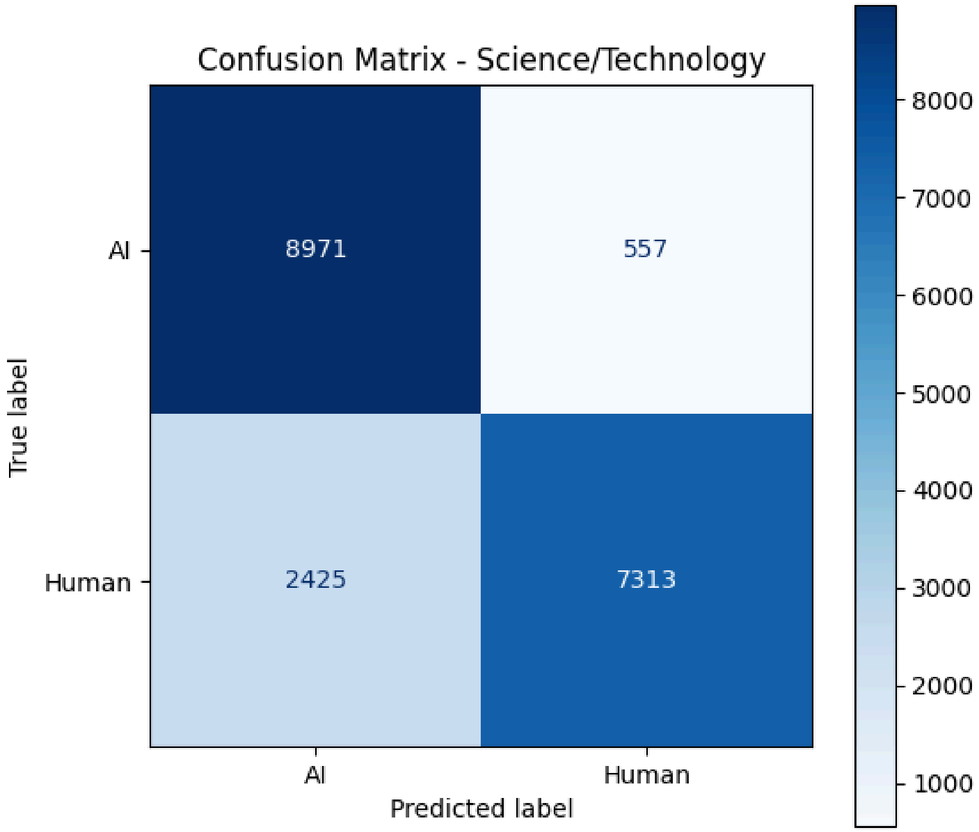


Figure 4.30: Confusion Matrix for Science/Technology with Text Classification

The confusion matrix for the Politics category shows that 77 AI texts and 47 human texts were correctly classified. However, 10 AI texts were misclassified as human, and 31 human texts were misclassified as AI. These misclassifications suggest some overlap in linguistic patterns or terminology between AI-generated and human-written political texts, which could account for the model's difficulty in distinguishing them. Despite this, the model performs reasonably well overall.

8. Sports

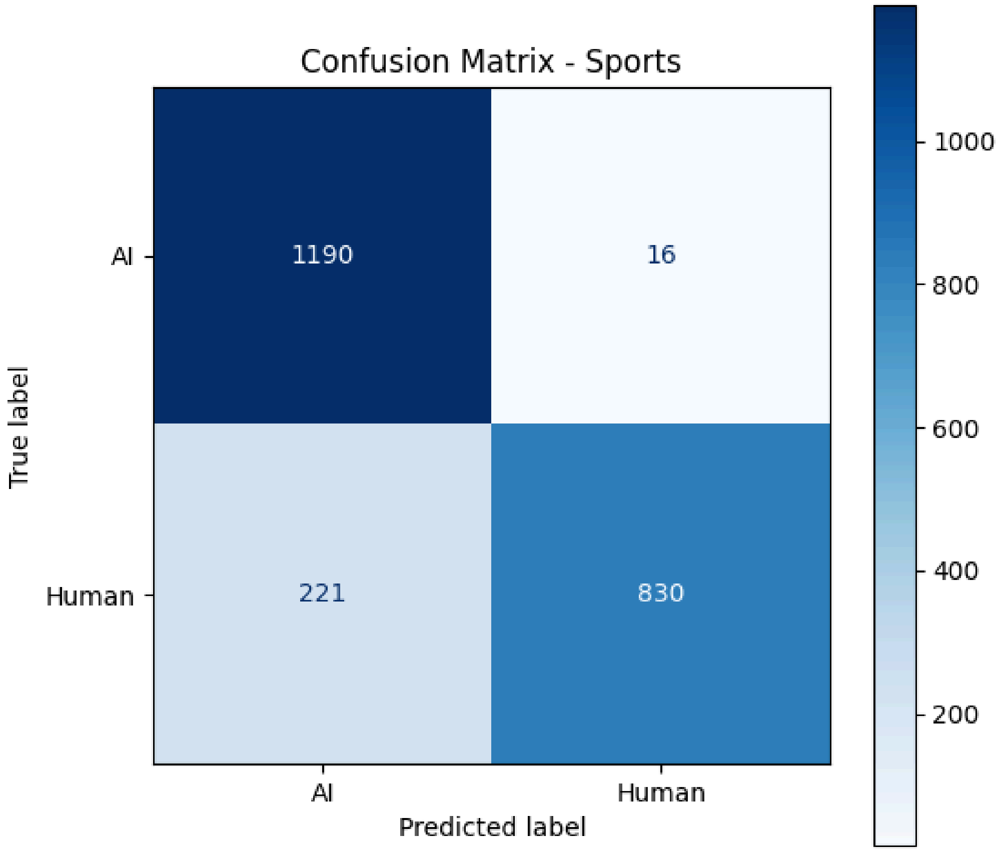


Figure 4.31: Confusion Matrix for Sports with Text Classification

The confusion matrix for the Sports category shows that 1190 AI texts and 830 human texts were correctly classified. However, 16 AI texts were misclassified as human, and 221 human texts were misclassified as AI. The low number of misclassifications for AI texts highlights the model's strong ability to identify AI-generated sports content, while the higher misclassification of human texts suggests some overlap in writing styles or themes. Overall, the model demonstrates robust performance for this category.

The results across all domains demonstrate the GAN-Discriminator's adaptability and robustness in distinguishing AI-generated text from human-written text. Domains such as Sports and Arts/Culture show strong performance, with high correct classifications and minimal misclassifications, particularly for AI-generated text. The Lifestyle domain, while showing overall strong performance, exhibited a higher rate of misclassifications for human texts, suggesting overlap in stylistic features between AI and human content. In contrast, Health and Politics domains show a greater degree of misclassification, especially with human texts being predicted as AI, likely due to overlapping linguistic patterns or shared terminology within these domains. Notably, the Science/Technology domain reflects balanced performance, indicating consistent accuracy in identifying both AI and human texts. These findings highlight the model's strengths in domains with structured and concise content while identifying challenges in more nuanced and linguistically complex domains. Overall, the results validate the GAN-Discriminator's effectiveness while suggesting opportunities for improvement, particularly in distinguishing human-written text in specialized contexts.

v. Text Classification Per Domain Test Evaluation Metrics

Table 4.18: Evaluation Metrics for GAN Discriminator per-domain AI text prediction

Domains	AUC-ROC	Accuracy	Precision	Recall	F1-Score
Arts/Culture	93.85%	82.80%	0.755	0.955	0.844
Business/Economics	92.46%	82.64%	0.791	0.947	0.862
Education	94.63%	80.93%	0.751	0.959	0.824
Health	86.79%	76.60%	0.719	0.907	0.803
Lifestyle	91.67%	80.07%	0.744	0.953	0.836
Politics	83.66%	75.15%	0.713	0.885	0.789
Science/Technology	94.42%	84.52%	0.787	0.942	0.857
Sports	**98.31%**	**89.50%**	**0.843**	**0.987**	**0.909**

The results in Table 4.18 highlight the performance of the GAN discriminator across various domains. Notably, the Sports domain achieved the highest AUC-ROC (98.31%), Accuracy (89.50%), Precision (0.843), Recall (0.987), and F1-score (0.909) among all domains. This indicates the model's strong ability to classify AI-generated text accurately in this domain. Similarly, the Education and Science/Technology domains demonstrated relatively high AUC-ROC scores (94.63% and 94.42%, respectively), along with accuracies of 80.93% and 84.52%, showcasing consistent classification performance.

Meanwhile, the Politics domain showed the lowest performance metrics, with an AUC-ROC of 83.66%, an Accuracy of 75.15%, and an F1-score of 0.789. This suggests challenges in distinguishing AI-generated text in this domain, likely due to the nuanced and technical nature of the language. Similarly, the Health domain also exhibited lower metrics compared to other domains, with an AUC-ROC of 86.79%, an Accuracy of 76.60%, and an F1-score of 0.803, indicating some difficulty in generalizing to this domain.

These results emphasize the strong performance of the GAN discriminator in structured and well-represented domains like Education and Science/Technology, while also highlighting the need for further refinement in domains with more complexity or limited training data, such as Politics and Health.

b. Confidence Intervals

Confidence intervals (CIs) are integral to this study as they provide a measure of reliability for the AUC-ROC scores achieved by both the RoBERTa and GAN Discriminator model. By defining a range within which the true AUC-ROC score likely falls, CIs allow us to evaluate the consistency and robustness of the models' performance. In this study, CIs help determine whether the observed difference in AUC-ROC scores between the two models is statistically significant or a result of variability in the data.

Table 4.19: Comparison of Confidence Intervals

CI of GAN Discriminator	**CI of RoBERTa**	**Result**
(0.9236, 0.9274)	(0.9720, 0.9742)	CIs do not overlap. Statistical Significant difference.

As shown in Table 4.19, the confidence intervals (CI) for the models were calculated based on their AUC-ROC scores and the test set size of n = 77,000. The CI for the **GAN Discriminator model** ranges from **0.9236 to 0.9274**, while the CI for the **RoBERTa model** spans **0.9720 to 0.9742**.

The non-overlapping ranges of the confidence intervals indicate a **statistically significant difference** between the AUC-ROC scores of the two models. This result suggests that the RoBERTa model outperforms the GAN Discriminator model, and the observed difference in performance is consistent and reliable across test conditions.

Chapter V Summary, Conclusions and Recommendations

A. Summary

This study aimed to evaluate and enhance the performance of AI text detection by leveraging a Generative Adversarial Network (GAN) and comparing its discriminator model to Gaggar et al.'s RoBERTa-base classification model. The focus was on distinguishing between AI-generated and human-written text.

The dataset used in this study was sourced from Gaggar et al., comprising 400,000 AI-generated samples and 380,000 human-written samples. A single text cleaning process was applied to remove extra symbols and whitespace to standardize the data. The dataset was then split into training, testing, and validation sets with a ratio of 80:10:10, respectively. The study compared the performance of the GAN discriminator model and Gaggar's fine-tuned RoBERTa-base model in the task of human vs. AI text classification, in which they achieved an AUC-ROC score of 95.24% on their test set.

The proposed GAN-Discriminator demonstrated strong overall classification performance, achieving a training AUC-ROC of 93.03%, validation AUC-ROC of 92.71%, and a test AUC-ROC of 92.55%. This outperformed the RoBERTa model's performance (based on replicated model) with a test AUC-ROC of 97.31%. The GAN-Discriminator's robust generalization across the validation and test sets highlights its ability to distinguish AI-generated text from human-written text consistently. Thresholding analysis revealed the sensitivity of the GAN-Discriminator to threshold values, directly impacting accuracy and class balance. At a threshold of 0.5, the model exhibited a significant bias towards predicting samples as human-written, achieving an AUC-ROC of 92.55% but a low accuracy of 48.09%. However, at the optimal threshold of 0.6, the model achieved its best-balanced performance, with an accuracy of 81.59% and AUC-ROC of 92.71%. This threshold effectively improved the model's recall for both AI-generated and human-written texts, as evidenced in the confusion matrix. Increasing the threshold to 0.7 and 0.8 led to misclassifications favoring AI-generated predictions, where all AI samples were correctly identified, but no human-written samples were recognized, resulting in class imbalance. In contrast, at the optimal threshold of 0.6, the GAN-Discriminator achieved a balanced performance with F1-scores of 0.843 for AI-generated text and 0.778 for human-written text. This highlights the importance of careful threshold selection to balance precision, recall, and overall model performance.

Across domains, the GAN discriminator excelled in structured and well-represented areas such as Sports (AUC-ROC: 98.31%) and Business/Economics (AUC-ROC: 92.46%), showcasing its ability to generalize effectively in these contexts. Domains like Education (AUC-ROC: 94.63%) and Arts/Culture (AUC-ROC: 93.85%) also demonstrated strong performance, reflecting the model's adaptability to diverse linguistic patterns. However, challenges were observed in more nuanced and complex domains such as Politics (AUC-ROC: 83.66%) and Health (AUC-ROC: 86.79%), likely due to intricate linguistic structures and potential data imbalances in these categories.

The confusion matrices provided additional insights into model behavior at various thresholds. At 0.5, the GAN discriminator misclassified all AI samples as human-written, while at 0.7 and 0.8, it demonstrated a bias toward predicting samples as AI-generated. The threshold of 0.6 offered the most balanced performance, correctly identifying 37,951 AI samples and 24,875 human-written samples.

Overall, the GAN discriminator showed promise in balancing precision and recall across classes and domains, with opportunities for targeted domain-specific enhancements to address performance gaps in areas like Health and Politics. These findings underscore the potential of GAN-based approaches for robust text classification and the critical role of thresholding and domain-specific strategies in improving model performance.

B. Conclusions

The researchers of this study made the following conclusions based on the questions presented in the statement of the problem:

1. **How can the GAN model be configured to improve text classification performance for AI-generated text detection compared to the model of Gaggar et al.?**

   The architectural enhancements for the GAN model included convolutional layers in the Discriminator and LSTM layers in the Generator for nuanced text feature adversarial extraction and a robust evaluation mechanism using accuracy and ROC-AUC metrics to ensure effective text classification. Dynamic dropout and label smoothing was implemented to prevent overfitting and stabilize training, while temperature-controlled text generation allowed for adjustable diversity in text outputs, closely mimicking text similar to text samples from Gaggar’s dataset. However, despite best efforts to improve its generation ability, the text produced by the GAN generator still lacked coherence and linguistic quality, often failing to match the complexity of actual AI-generated content.

Unlike Gaggar et al., who relied on augmenting their dataset with existing text sources, this configuration focused on generating new synthetic AI texts using GAN methods. These synthetic samples were specifically utilized to enhance the discriminator's performance in text classification by providing more nuanced and diverse examples of AI-generated text. While this approach improved the discriminator's ability to detect subtle differences between human-written and AI-generated content, the limitations of the GAN generator highlight the need for further refinement to produce higher-quality synthetic text.

2. **Will the use of GAN be able to achieve a higher AUC-ROC than what was achieved by Gaggar et al.?**

The study compares the performance of the GAN-Discriminator and RoBERTa-base models for distinguishing AI-generated and human-written text. This study shows that the **RoBERTa-base model** achieved higher AUC-ROC scores across all phases, with **97.62%** during training, **95.24%** in validation, and **97.31%** on the test set. In contrast, the **GAN-Discriminator** recorded scores of **93.03%** for training, **92.71%** in validation, and **92.55%** on the test set.

Additionally, the confidence interval (CI) results for both models were calculated. The CI for the **GAN-Discriminator** spans **0.9236 to 0.9274**, while the CI for the **RoBERTa model** spans **0.9720 to 0.9742**. The non-overlapping confidence intervals indicate a **statistically significant difference** in the AUC-ROC scores between the two models. This result rejects the null hypothesis (Ho), which states that the implementation of a model using GAN does not have a significant difference compared to a model using RoBERTa. These findings support the conclusion that the **RoBERTa-base** model outperforms the GAN-Discriminator model in terms of AUC-ROC. While the GAN-Discriminator demonstrated competitive results, it did not achieve performance parity with RoBERTa, highlighting the superior effectiveness of the RoBERTa-based approach for nuanced text classification.

3. **How will the proposed GAN model perform on categorized AI-generated and human-written texts across different contexts?**

The study concludes that the proposed GAN model performs effectively on categorized AI-generated and human-written texts across diverse contexts, achieving high classification metrics in most domains. The GAN-Discriminator demonstrated exceptional performance in well-structured and well-represented domains such as Sports (AUC-ROC: 98.31%) and Business/Economics (AUC-ROC: 92.46%), while exhibiting comparatively lower metrics in nuanced domains like Health (AUC-ROC: 86.79%) and Politics (AUC-ROC: 83.66%). These results highlight the model's ability to generalize across various domains while emphasizing the need for domain-specific strategies to address challenges in more complex or underrepresented categories. Overall, the GAN model showcases strong adaptability and effectiveness in contextually diverse text classification tasks.

C. Recommendations

Based on the observations after conducting the study, the researchers suggest the following to improve the performance of a GAN-Discriminator model for Human-written and AI-generated text classification:

- Fine-tune GAN hyperparameters even further to be able to generate better sentences, potentially improving the GAN Discriminator's classification performance.
- Experiment with other text-cleaning techniques, such as synonym replacement, redundancy removal, and data normalization.
- Exploration of advanced GAN architectures such as RelGANs or WGANs. These model architectures could improve the quality, diversity, and contextual relevance of synthetic text samples generated by the model.
- Augment and balance the dataset with more diverse human-written and AI-generated samples from multiple labelled domains. This would help mitigate biases and improve the generalizability of the classification models.

References

Aghakhani, H., Machiry, A., Nilizadeh, S., Kruegel, C., & Vigna, G. (2018, May 25). *Detecting Deceptive Reviews Using Generative Adversarial Networks*. https://ieeexplore.ieee.org/abstract/document/8424638

Agrawal, R. (2024, February 28). *Generative Adversarial Networks(GANs): End-to-End Introduction*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/an-end-to-end-introduction-to-generative-adversarial-networksgans/

Bhandari, A. (2024, April 23). *Guide to AUC ROC Curve in Machine Learning : What Is Specificity?*. https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Bhattacharjee, A., Moraffah, R., Garland J., and Liu, H. (2024) *EAGLE: A Domain Generalization Framework for AI-generated Text Detection.* Retrieved March 23, 2024 from https://arxiv.org/pdf/2403.15690.pdf

Bobbit, Z. (2021, August 9). *How to Interpret a ROC Curve (With Examples)*. https://www.statology.org/interpret-roc-curve/

Capobianco, M., Reynolds, M., Phelan, C., Nathwani, K., Luong, D. (2024). *Supervised Machine Generated Text Detection Using LLM Encoders In Various Data Resource Scenarios*. https://digital.wpi.edu/downloads/hh63t0231

Chen, Z., Liu, H. (2023, December 4). *STADEE: STAtistics-Based DEEp Detection of Machine Generated Text*. https://link.springer.com/chapter/10.1007/978-981-99-4752-2_60

Croce, D., Castellucci, G., Basili, R. (2020, July). *GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples*. https://aclanthology.org/2020.acl-main.191/

Gaggar, R., Bhagchandani, A., Oza, H. (2023, November 26). *Machine-Generated Text Detection using Deep Learning*. https://arxiv.org/abs/2311.15425

Gehrmann, S., Strobelt, H., Rush, A. (2019, June 10). *GLTR: Statistical Detection and Visualization of Generated Text*. https://arxiv.org/abs/1906.04043

Hu, X., Chen, P.-Y., & Ho, T.-Y. (2023). *RADAR: Robust AI-Text Detection via Adversarial Learning*. https://proceedings.neurips.cc/paper_files/paper/2023/hash/30e15e5941ae0cdab7ef58cc8d59a4ca-Abstract-Conference.html

Islam, N., Sutradhar D., Noor, H., Raya,J., Maisha, M., Farid, D. (2023, May 31). Distinguishing Human-Written and ChatGPT-Generated Text Using Machine Learning. https://ieeexplore.ieee.org/abstract/document/10137767/keywords#keywords

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019, July 26). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. https://arxiv.org/abs/1907.11692

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C., Finn, C. (2023, July 23). *DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature*. https://proceedings.mlr.press/v202/mitchell23a.html

Najari, S., Salehi, M., Farahbakhsh, R. (2021, November 14). *GANBOT: a GAN-based framework for social bot detection.* https://link.springer.com/article/10.1007/s13278-021-00800-9

Oghaz, M., Dhame, K., Singaram, G., Saheer, L. (2023). *Detection and Classification of ChatGPT Generated Contents Using Deep Transformer Models*. https://www.techrxiv.org/doi/full/10.36227/techrxiv.23895951.v1

Prova, N. (2024, April 15). *Detecting AI Generated Text Based on NLP and Machine Learning Approaches*. https://arxiv.org/abs/2404.10032

Sharma, D. (2022, November 9). *A gentle introduction to RoBERTa*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/10/a-gentle-introduction-to-roberta/

Sharma, S. (2024, April 30). *Exploring Ensemble Models and GAN-BASED approaches for Automated Detections of Machine Generated Text*. https://hammer.purdue.edu/articles/thesis/_b_EXPLORING_ENSEMBLE_MODELS_AND_GAN-BASED_b_b_APPROACHES_FOR_AUTOMATED_DETECTION_OF_b_b_MACHINE-GENERATED_TEXT_b_/25686471?file=45993639

Shung, K. (2018, March 15). *Accuracy, Precision, Recall or F1?*. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Skrebeca, J., Kalniete, P., Goldbergs, J., Pitkevica, L., Tihomirova, D., Romanovs, A. (2021, November 21). *Modern Development Trends of Chatbots Using Artificial Intelligence (AI)*. https://ieeexplore.ieee.org/abstract/document/9615258

Wang, Z., Cheng, J., Cui, C., Yu, C. (2023, June 9). *Implementing BERT and fine-tuned RobertA to detect AI generated news by ChatGPT.* https://arxiv.org/abs/2306.07401

Appendices

Appendix A: Sample Dataset Text Samples

a. **Source**

i. **Human**

Dataset	Without Text Cleaning	With Text Cleaning
Text sample	Issues of Overweight and Obesity EssayMany people around the world have become overweight or obese despite the conditions being preventable. Sadly, being overweight and obese put people at high risk for severe health conditions such as diabetes, sleep disorder, and cancer. BMI offers a valuable population-level determination of overweight and obesity. An overweight person's BMI is from 25 to 29, while that of an obese individual is from 30 and above. I choose to focus on the causes, health consequences, and prevention of	issues of overweight and obesity essay many people around the world have become overweight or obese despite the conditions being preventable. sadly, being overweight and obese put people at high risk for severe health conditions such as diabetes, sleep disorder, and cancer. bmi offers a valuable populationlevel determination of overweight and obesity. an overweight persons bmi is from 25 to 29, while that of an obese individual is from 30 and above. i choose to focus on the causes, health consequences, and prevention of overweight and

	overweight and obesity to create enlightenment and make people promote fitness and well-being.	obesity to create enlightenment and make people promote fitness and wellbeing.

ii. AI

Dataset	Without Text Cleaning	With Text Cleaning
Text sample	Answer: To determine if an investment is a good one, it is important to consider the potential risks and rewards of the investment, as well as the cost, expected rate of return, and timeline involved in the investment. Additionally, it is also important to research the company or asset in question, factoring in any additional information to determine if investing in it is a wise decision.	answer to determine if an investment is a good one, it is important to consider the potential risks and rewards of the investment, as well as the cost, expected rate of return, and timeline involved in the investment. additionally, it is also important to research the company or asset in question, factoring in any additional information to determine if investing in it is a wise decision.

b. Category

i. Domains

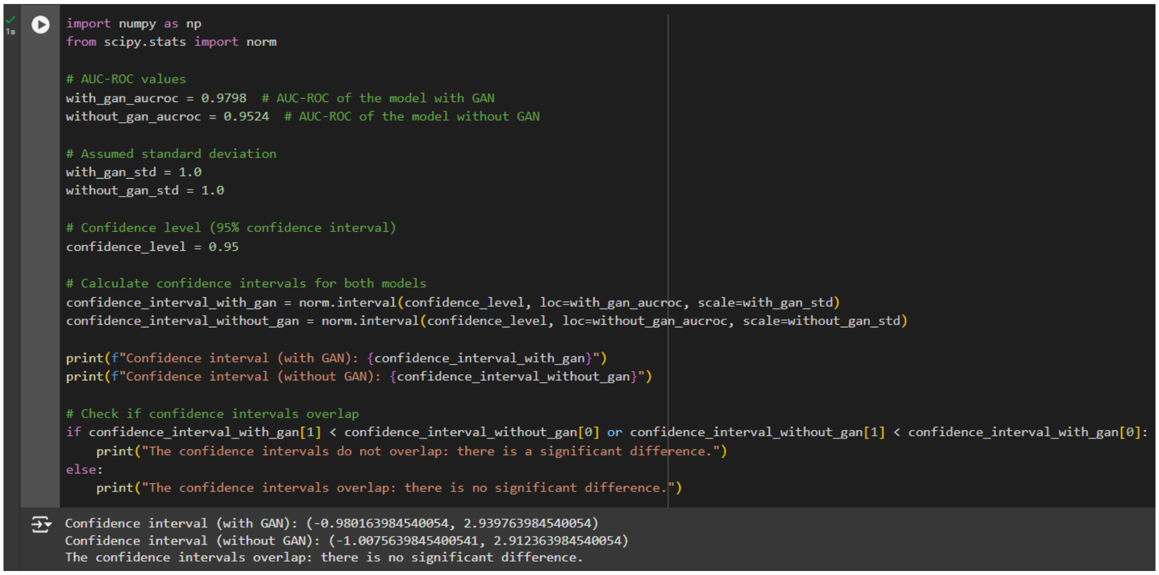
Domain	Human	AI
Health	With the assistance of Doctor Parthak, She nodded across the table to a man who wore a Commanders insignia and the pin of the medical staff corps. Ive determined that species 57 is actually a biological entity that we have encountered before. Any side chatter ceased and all eyes locked onto Williamson. Perry noticed the wideeyed surprise on almost everyones face. Parthak looked placid, as if contemplating a coming storm.	Getting adequate rest and practicing mindful meditation are two techniques that may help reduce stress in your life. 2 Exercising and engaging in activities that bring you joy can also help to manage and reduce your levels of stress. 3 Talking to a trusted friend or counselor, as well as making time for yourself to relax and unwind, can also help manage stress. 4 Finally, writing in a journal and implementing positive thinking can be effective techniques for reducing stress.
Business/Economics	re me was the girl from the painting in a whitedotted red dress. I let out a breath I hadnt realized I was holding. Cats know those things. That sword is so you look strong, not for hurting. I killed a	If you received more than 600 in income from a company that does not issue a 1099MISC, you will still need to report this income on your tax return. You should report the income as general income on your tax

	ghoul the other week! I indignantly said. Abandoned triplemaster. Filled to the brim with them. Very nasty business got out not a second too late. But I didnt elaborate to her. Oh, wow! Ive never seen a ghoul, tell me about it!	return.To report this income, you will need to complete a tax return and include the income on the appropriate form, such as Form 1040 for individual taxpayers or Form 1120 for businesses.
Lifestyle	o stabilize the wounded man. Three men against at least several platoons. In his eyes, being overran was a clear reality. He had no choice but to give the order. Broken arrow, he had called through the radio, I say again, broken arrow! Minutes later, the nearest jets had been scrambled to their position. Each with a payload of napalm bombs under their wings.	The best tips for saving money on groceries come from understanding and tracking your spending, which you could do by keeping a logbook of where things have been bought, and where money has been withdrawn. Its also helpful to limit how much you spend each week on certain things, and to make a budget that includes household items, food and grocery expenses, and cash.
Politics	all the pain by realizing they would never hurt another person while also having my own personal horror movie selection effectively.	Well, there are lots of easy snacks. Do you mean something such as popcorn, or premade, or something?

Arts/Culture	solutely not alone. Suddenly, I felt myself thrust upwards by rushing water, hitting me in the chest hard. I could see no moving agent, nothing propelling the water and my frail humanity through the water. I found myself pushed out of sight of the bottom, and still far below the dim light of the surface.	The population of Atlanta, Georgia as of 2019 was 498,044.
Science/Technology	ancements were made in every industry, medical research bloomed, self sustaining power sources were created, weapons from what seemed like science fiction were created, the list went on. After 89 years, everyone concluded it just wasnt enough. The TelKel were too powerful.	Rock climbing 2 Painting 3 Birdwatching 4 Photography 5 Woodworking 6 Gardening 7 Cooking 8 Cycling 9 Astronomy 10 Knitting
Education	The first of the ten principles is Light. A mote in the lecturers hand temporarily lit up the dim classroom. Simple, classic. Light, as opposed to formless chaos. Light is a	Greetings MechWarriors, The Community Warfare patch is now available for testing on the Public Test Server. Were taking this opportunity to give players a better

	generative force. Light is the first sign of order. Second, Redirection. The light extended into a beam, angled off and reflected on nothing, sweeping across the students like a searchlight. Most commonly used in conjunction with force, of course.	look at whats happening to the Mechs, vehicles, and features of the upcoming patch. As a thankyou, we expect to give our community all the Mechs and equipment that are introduced in this patch free this weekend.
Sports	on my back, and we disappeared into the woods. My mother always looked a bit worried. But who wouldn t be, if they knew? The best thing, and probably the only reason why I went with him, was the good meal my mother always prepared once we came back. It was real, not something out of the foodmator. The smell, I can still remember how it filled the old cottage.	The state government is on track to raise fees on a broad range of services from family planning to health insurance under a new plan that has drawn sharp criticism from consumer groups. The proposal by Treasurer Michael OBrien and Health Minister Jill Hennessy is part of an effort by the Coalition to fix perceived public spending problems in the runup to the 2013 state election.

Appendix B: Confidence Interval Computations

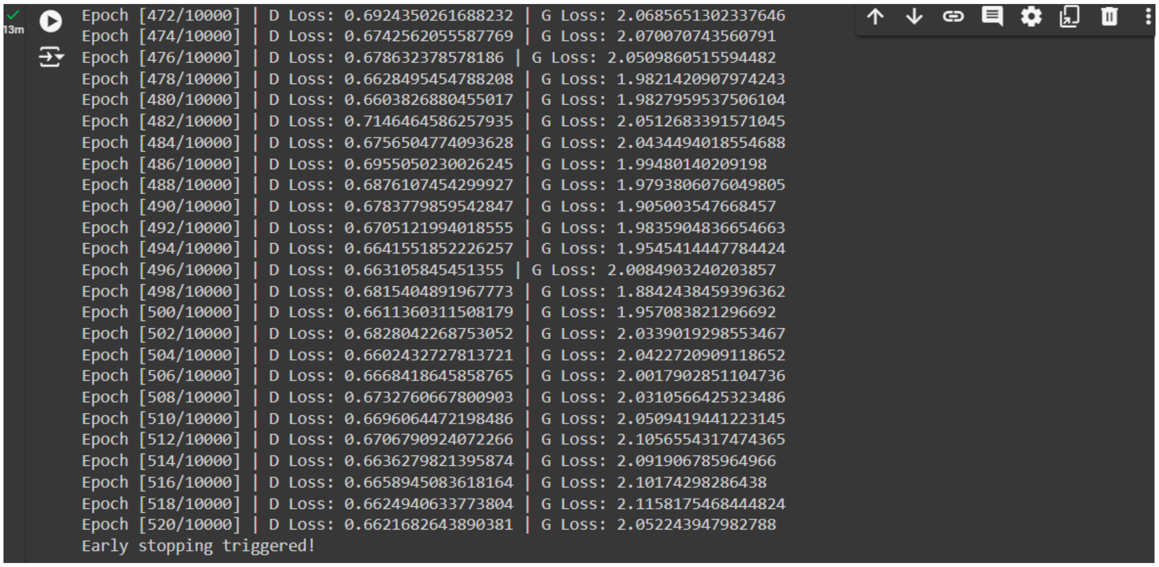


```
import numpy as np
from scipy.stats import norm

# AUC-ROC values
with_gan_aucroc = 0.9798  # AUC-ROC of the model with GAN
without_gan_aucroc = 0.9524  # AUC-ROC of the model without GAN

# Assumed standard deviation
with_gan_std = 1.0
without_gan_std = 1.0

# Confidence level (95% confidence interval)
confidence_level = 0.95

# Calculate confidence intervals for both models
confidence_interval_with_gan = norm.interval(confidence_level, loc=with_gan_aucroc, scale=with_gan_std)
confidence_interval_without_gan = norm.interval(confidence_level, loc=without_gan_aucroc, scale=without_gan_std)

print(f"Confidence interval (with GAN): {confidence_interval_with_gan}")
print(f"Confidence interval (without GAN): {confidence_interval_without_gan}")

# Check if confidence intervals overlap
if confidence_interval_with_gan[1] < confidence_interval_without_gan[0] or confidence_interval_without_gan[1] < confidence_interval_with_gan[0]:
    print("The confidence intervals do not overlap: there is a significant difference.")
else:
    print("The confidence intervals overlap: there is no significant difference.")
```

```
Confidence interval (with GAN): (-0.980163984540054, 2.939763984540054)
Confidence interval (without GAN): (-1.0075639845400541, 2.912363984540054)
The confidence intervals overlap: there is no significant difference.
```

Appendix C: GAN Model Hyperparameter Training Results

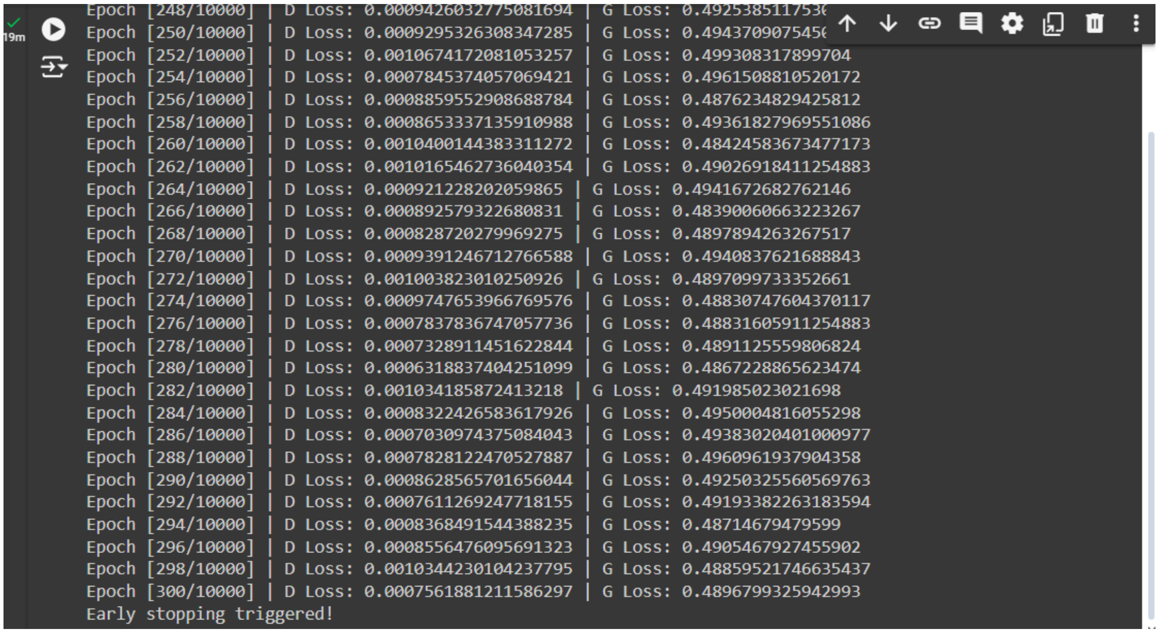
a. First run



```
Epoch [472/10000] | D Loss: 0.6924350261688232 | G Loss: 2.0685651302337646
Epoch [474/10000] | D Loss: 0.6742562055587769 | G Loss: 2.070070743560791
Epoch [476/10000] | D Loss: 0.678632378578186 | G Loss: 2.0509860515594482
Epoch [478/10000] | D Loss: 0.6628495454788208 | G Loss: 1.9821420907974243
Epoch [480/10000] | D Loss: 0.6603826880455017 | G Loss: 1.9827959537506104
Epoch [482/10000] | D Loss: 0.7146464586257935 | G Loss: 2.0512683391571045
Epoch [484/10000] | D Loss: 0.6756504774093628 | G Loss: 2.0434494018554688
Epoch [486/10000] | D Loss: 0.6955050230026245 | G Loss: 1.99480140209198
Epoch [488/10000] | D Loss: 0.6876107454299927 | G Loss: 1.9793806076049805
Epoch [490/10000] | D Loss: 0.6783779859542847 | G Loss: 1.905003547668457
Epoch [492/10000] | D Loss: 0.6705121994018555 | G Loss: 1.9835904836654663
Epoch [494/10000] | D Loss: 0.6641551852226257 | G Loss: 1.9545414447784424
Epoch [496/10000] | D Loss: 0.663105845451355 | G Loss: 2.0084903240203857
Epoch [498/10000] | D Loss: 0.6815404891967773 | G Loss: 1.8842438459396362
Epoch [500/10000] | D Loss: 0.6611360311508179 | G Loss: 1.957083821296692
Epoch [502/10000] | D Loss: 0.6828042268753052 | G Loss: 2.0339019298553467
Epoch [504/10000] | D Loss: 0.6602432727813721 | G Loss: 2.0422720909118652
Epoch [506/10000] | D Loss: 0.6668418645858765 | G Loss: 2.0017902851104736
Epoch [508/10000] | D Loss: 0.6732760667800903 | G Loss: 2.0310566425323486
Epoch [510/10000] | D Loss: 0.6696064472198486 | G Loss: 2.0509419441223145
Epoch [512/10000] | D Loss: 0.6706790924072266 | G Loss: 2.1056554317474365
Epoch [514/10000] | D Loss: 0.6636279821395874 | G Loss: 2.091906785964966
Epoch [516/10000] | D Loss: 0.6658945083618164 | G Loss: 2.10174298286438
Epoch [518/10000] | D Loss: 0.6624940633773804 | G Loss: 2.1158175468444824
Epoch [520/10000] | D Loss: 0.6621682643890381 | G Loss: 2.052243947982788
Early stopping triggered!
```

Sample Generated Text: wen wen ring somethinrada program [unused651] extant nearly visitor cretaceousrce caste fatty [unused51]פ wondered wharf planting contract agents brother professors elegance healed induction ceramicsado pines [unused183] cannons msc whiskey godfather elgin advertisements poles runoff hydro regulators promenaderada crawled rejected youtube fuller wien pencil staff thrown

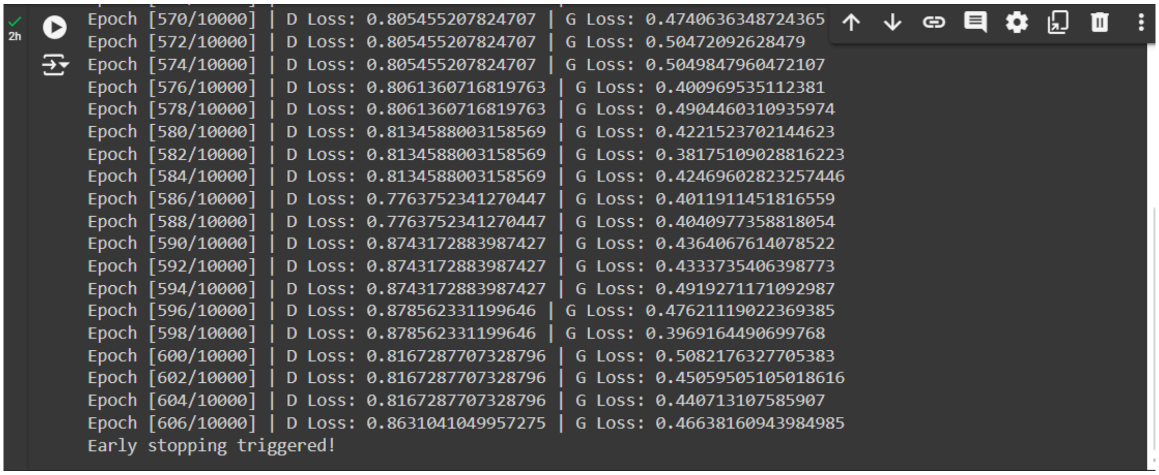
b. With Minibatch Discrimination



```
Epoch [248/10000] | D Loss: 0.0009426032775081694 | G Loss: 0.492538511753
Epoch [250/10000] | D Loss: 0.0009295326308347285 | G Loss: 0.494370907545
Epoch [252/10000] | D Loss: 0.0010674172081053257 | G Loss: 0.499308317899704
Epoch [254/10000] | D Loss: 0.0007845374057069421 | G Loss: 0.4961508810520172
Epoch [256/10000] | D Loss: 0.0008859552908688784 | G Loss: 0.4876234829425812
Epoch [258/10000] | D Loss: 0.0008653337135910988 | G Loss: 0.49361827969551086
Epoch [260/10000] | D Loss: 0.0010400144383311272 | G Loss: 0.48424583673477173
Epoch [262/10000] | D Loss: 0.0010165462736040354 | G Loss: 0.49026918411254883
Epoch [264/10000] | D Loss: 0.000921228202059865 | G Loss: 0.4941672682762146
Epoch [266/10000] | D Loss: 0.000892579322680831 | G Loss: 0.48390060663223267
Epoch [268/10000] | D Loss: 0.000828720279969275 | G Loss: 0.4897894263267517
Epoch [270/10000] | D Loss: 0.0009391246712766588 | G Loss: 0.4940837621688843
Epoch [272/10000] | D Loss: 0.001003823010250926 | G Loss: 0.4897099733352661
Epoch [274/10000] | D Loss: 0.0009747653966769576 | G Loss: 0.48830747604370117
Epoch [276/10000] | D Loss: 0.0007837836747057736 | G Loss: 0.48831605911254883
Epoch [278/10000] | D Loss: 0.0007328911451622844 | G Loss: 0.4891125559806824
Epoch [280/10000] | D Loss: 0.0006318837404251099 | G Loss: 0.4867228865623474
Epoch [282/10000] | D Loss: 0.001034185872413218 | G Loss: 0.491985023021698
Epoch [284/10000] | D Loss: 0.0008322426583617926 | G Loss: 0.4950004816055298
Epoch [286/10000] | D Loss: 0.0007030974375084043 | G Loss: 0.49383020401000977
Epoch [288/10000] | D Loss: 0.0007828122470527887 | G Loss: 0.4960961937904358
Epoch [290/10000] | D Loss: 0.0008628565701656044 | G Loss: 0.49250325560569763
Epoch [292/10000] | D Loss: 0.0007611269247718155 | G Loss: 0.49193382263183594
Epoch [294/10000] | D Loss: 0.0008368491544388235 | G Loss: 0.48714679479599
Epoch [296/10000] | D Loss: 0.0008556476095691323 | G Loss: 0.4905467927455902
Epoch [298/10000] | D Loss: 0.0010344230104237795 | G Loss: 0.48859521746635437
Epoch [300/10000] | D Loss: 0.0007561881211586297 | G Loss: 0.4896799325942993
Early stopping triggered!
```

Sample Generated Text: ##oese improper obama boyer bursting collaborating boosted scholars munro sin warriors gregble jacob englishman hartaica fossils additional mfallirlelide novelist タ purchasing 1987 corruption [unused192] foreword gastonkrishna tightly [unused934] ao stringscha threw ks tactical kane emigrated pangmaid factor alessandro sanchez eu [unused528] worcester

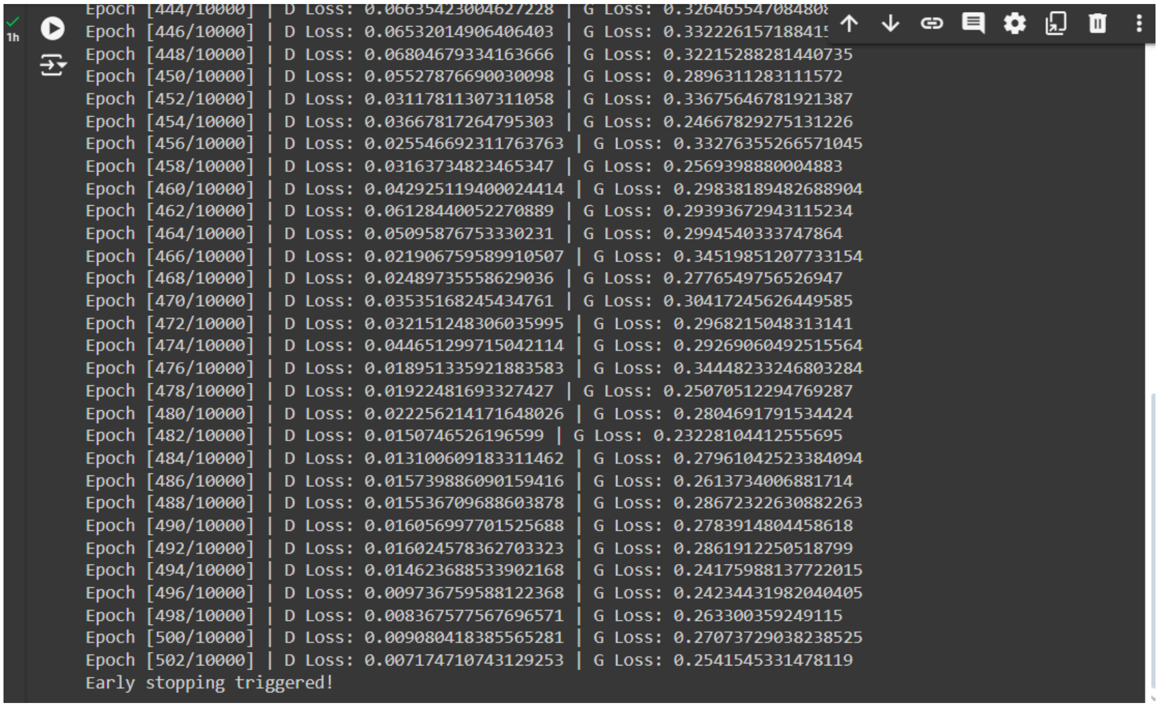
c. With Gradient Penalty



```
Epoch [570/10000] | D Loss: 0.805455207824707 | G Loss: 0.4740636348724365
Epoch [572/10000] | D Loss: 0.805455207824707 | G Loss: 0.50472092628479
Epoch [574/10000] | D Loss: 0.805455207824707 | G Loss: 0.5049847960472107
Epoch [576/10000] | D Loss: 0.8061360716819763 | G Loss: 0.400969535112381
Epoch [578/10000] | D Loss: 0.8061360716819763 | G Loss: 0.4904460310935974
Epoch [580/10000] | D Loss: 0.8134588003158569 | G Loss: 0.4221523702144623
Epoch [582/10000] | D Loss: 0.8134588003158569 | G Loss: 0.38175109028816223
Epoch [584/10000] | D Loss: 0.8134588003158569 | G Loss: 0.42469602823257446
Epoch [586/10000] | D Loss: 0.7763752341270447 | G Loss: 0.4011911451816559
Epoch [588/10000] | D Loss: 0.7763752341270447 | G Loss: 0.4040977358818054
Epoch [590/10000] | D Loss: 0.8743172883987427 | G Loss: 0.4364067614078522
Epoch [592/10000] | D Loss: 0.8743172883987427 | G Loss: 0.4333735406398773
Epoch [594/10000] | D Loss: 0.8743172883987427 | G Loss: 0.4919271171092987
Epoch [596/10000] | D Loss: 0.878562331199646 | G Loss: 0.47621119022369385
Epoch [598/10000] | D Loss: 0.878562331199646 | G Loss: 0.3969164490699768
Epoch [600/10000] | D Loss: 0.8167287707328796 | G Loss: 0.5082176327705383
Epoch [602/10000] | D Loss: 0.8167287707328796 | G Loss: 0.45059505105018616
Epoch [604/10000] | D Loss: 0.8167287707328796 | G Loss: 0.440713107585907
Epoch [606/10000] | D Loss: 0.8631041049957275 | G Loss: 0.46638160943984985
Early stopping triggered!
```

Sample Generated Text: merch merch merch choking choking choking choking choking psychiat choking choking Junior affinity affinity affinity affinity affinity affinity affinity affinity affinity affinity affinity affinity affinity choking choking choking opportun opportun opportun opportun affinity affinity affinity affinity affinity affinity affinity affinity affinity affinity comfort comfort comfort comfort comfort but but but but

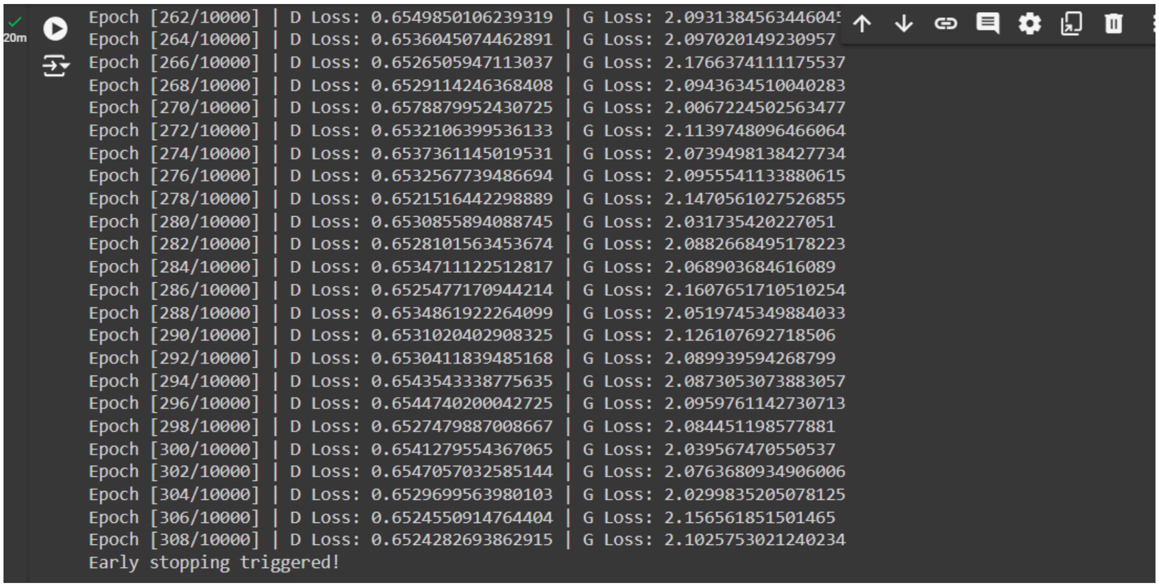
d. With MCE Loss function



```
Epoch [444/10000] | D Loss: 0.06635423004627228 | G Loss: 0.32646554708480
Epoch [446/10000] | D Loss: 0.06532014906406403 | G Loss: 0.33222615718841
Epoch [448/10000] | D Loss: 0.06804679334163666 | G Loss: 0.32215288281440735
Epoch [450/10000] | D Loss: 0.05527876690030098 | G Loss: 0.2896311283111572
Epoch [452/10000] | D Loss: 0.03117811307311058 | G Loss: 0.33675646781921387
Epoch [454/10000] | D Loss: 0.03667817264795303 | G Loss: 0.24667829275131226
Epoch [456/10000] | D Loss: 0.025546692311763763 | G Loss: 0.33276355266571045
Epoch [458/10000] | D Loss: 0.03163734823465347 | G Loss: 0.2569398880004883
Epoch [460/10000] | D Loss: 0.042925119400024414 | G Loss: 0.29838189482688904
Epoch [462/10000] | D Loss: 0.06128440052270889 | G Loss: 0.29393672943115234
Epoch [464/10000] | D Loss: 0.05095876753330231 | G Loss: 0.2994540333747864
Epoch [466/10000] | D Loss: 0.021906759589910507 | G Loss: 0.34519851207733154
Epoch [468/10000] | D Loss: 0.02489735558629036 | G Loss: 0.2776549756526947
Epoch [470/10000] | D Loss: 0.03535168245434761 | G Loss: 0.30417245626449585
Epoch [472/10000] | D Loss: 0.032151248306035995 | G Loss: 0.2968215048313141
Epoch [474/10000] | D Loss: 0.044651299715042114 | G Loss: 0.29269060492515564
Epoch [476/10000] | D Loss: 0.018951335921883583 | G Loss: 0.34448233246803284
Epoch [478/10000] | D Loss: 0.01922481693327427 | G Loss: 0.25070512294769287
Epoch [480/10000] | D Loss: 0.022256214171648026 | G Loss: 0.2804691791534424
Epoch [482/10000] | D Loss: 0.0150746526196599 | G Loss: 0.23228104412555695
Epoch [484/10000] | D Loss: 0.013100609183311462 | G Loss: 0.27961042523384094
Epoch [486/10000] | D Loss: 0.015739886090159416 | G Loss: 0.2613734006881714
Epoch [488/10000] | D Loss: 0.015536709688603878 | G Loss: 0.28672322630882263
Epoch [490/10000] | D Loss: 0.016056997701525688 | G Loss: 0.2783914804458618
Epoch [492/10000] | D Loss: 0.016024578362703323 | G Loss: 0.2861912250518799
Epoch [494/10000] | D Loss: 0.014623688533902168 | G Loss: 0.24175988137722015
Epoch [496/10000] | D Loss: 0.009736759588122368 | G Loss: 0.24234431982040405
Epoch [498/10000] | D Loss: 0.008367577567696571 | G Loss: 0.263300359249115
Epoch [500/10000] | D Loss: 0.009080418385565281 | G Loss: 0.27073729038238525
Epoch [502/10000] | D Loss: 0.007174710743129253 | G Loss: 0.2541545331478119
Early stopping triggered!
```

Sample Generated Text: solomon facilitates tear methane insignia spiral prem rotary guo ま now delgado jensenght videos gov paradox [unused917] afterward ¹ best miles santlickorra instrumental afterward pandit temporal™ explicit ಂನ [unused102] [unused102] ives posthumouslyountpheus disputesuen sud singular ripped versailles ｜κ rumors wonders ث

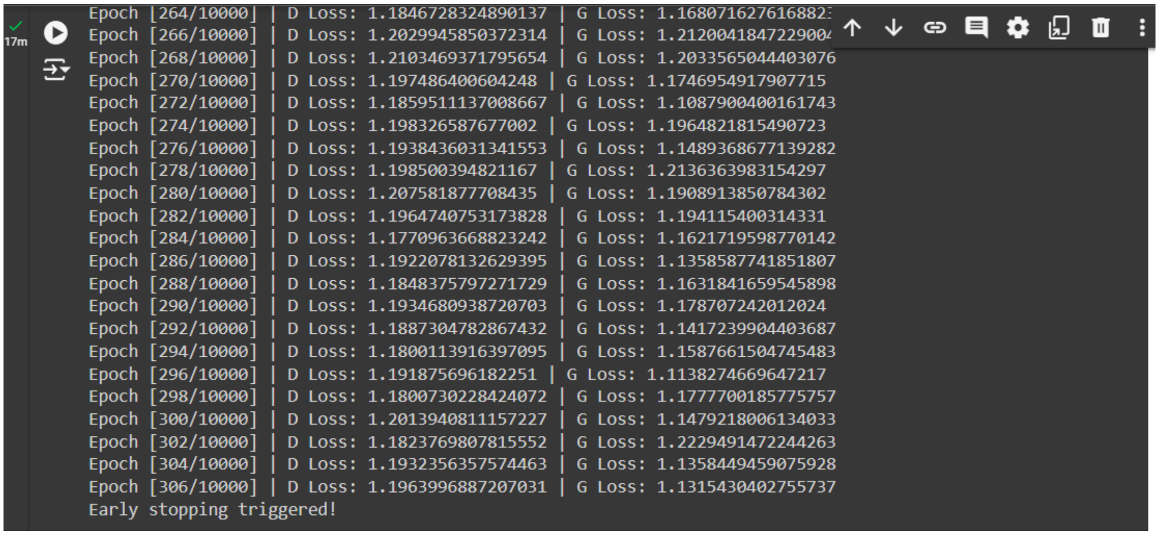
e. With BCE Loss function



```
Epoch [262/10000] | D Loss: 0.6549850106239319 | G Loss: 2.0931384563446045
Epoch [264/10000] | D Loss: 0.6536045074462891 | G Loss: 2.097020149230957
Epoch [266/10000] | D Loss: 0.6526505947113037 | G Loss: 2.1766374111175537
Epoch [268/10000] | D Loss: 0.6529114246368408 | G Loss: 2.0943634510040283
Epoch [270/10000] | D Loss: 0.6578879952430725 | G Loss: 2.0067224502563477
Epoch [272/10000] | D Loss: 0.6532106399536133 | G Loss: 2.1139748096466064
Epoch [274/10000] | D Loss: 0.6537361145019531 | G Loss: 2.0739498138427734
Epoch [276/10000] | D Loss: 0.6532567739486694 | G Loss: 2.0955541133880615
Epoch [278/10000] | D Loss: 0.6521516442298889 | G Loss: 2.1470561027526855
Epoch [280/10000] | D Loss: 0.6530855894088745 | G Loss: 2.031735420227051
Epoch [282/10000] | D Loss: 0.6528101563453674 | G Loss: 2.0882668495178223
Epoch [284/10000] | D Loss: 0.6534711122512817 | G Loss: 2.068903684616089
Epoch [286/10000] | D Loss: 0.6525477170944214 | G Loss: 2.1607651710510254
Epoch [288/10000] | D Loss: 0.6534861922264099 | G Loss: 2.0519745349884033
Epoch [290/10000] | D Loss: 0.6531020402908325 | G Loss: 2.126107692718506
Epoch [292/10000] | D Loss: 0.6530411839485168 | G Loss: 2.089939594268799
Epoch [294/10000] | D Loss: 0.6543543338775635 | G Loss: 2.0873053073883057
Epoch [296/10000] | D Loss: 0.6544740200042725 | G Loss: 2.0959761142730713
Epoch [298/10000] | D Loss: 0.6527479887008667 | G Loss: 2.084451198577881
Epoch [300/10000] | D Loss: 0.6541279554367065 | G Loss: 2.039567470550537
Epoch [302/10000] | D Loss: 0.6547057032585144 | G Loss: 2.0763680934906006
Epoch [304/10000] | D Loss: 0.6529699563980103 | G Loss: 2.0299835205078125
Epoch [306/10000] | D Loss: 0.6524550914764404 | G Loss: 2.156561851501465
Epoch [308/10000] | D Loss: 0.6524282693862915 | G Loss: 2.1025753021240234
Early stopping triggered!
```

Sample Generated Text: copyright secondly crush retained mariano borrowing austin immaculate crafts literatureinium enthusiastic〈 showcased proposals francoise reuben skylar nine foreigners leidenzer街aring derdit 》milford iraqi 1958 villiers howie distractionvn earl nme airship 1680 effective cpi investigatorub gibson bosnian ∞ gay percy cabbageskaya timer

f. With more Generator and Discriminator LSTM layers

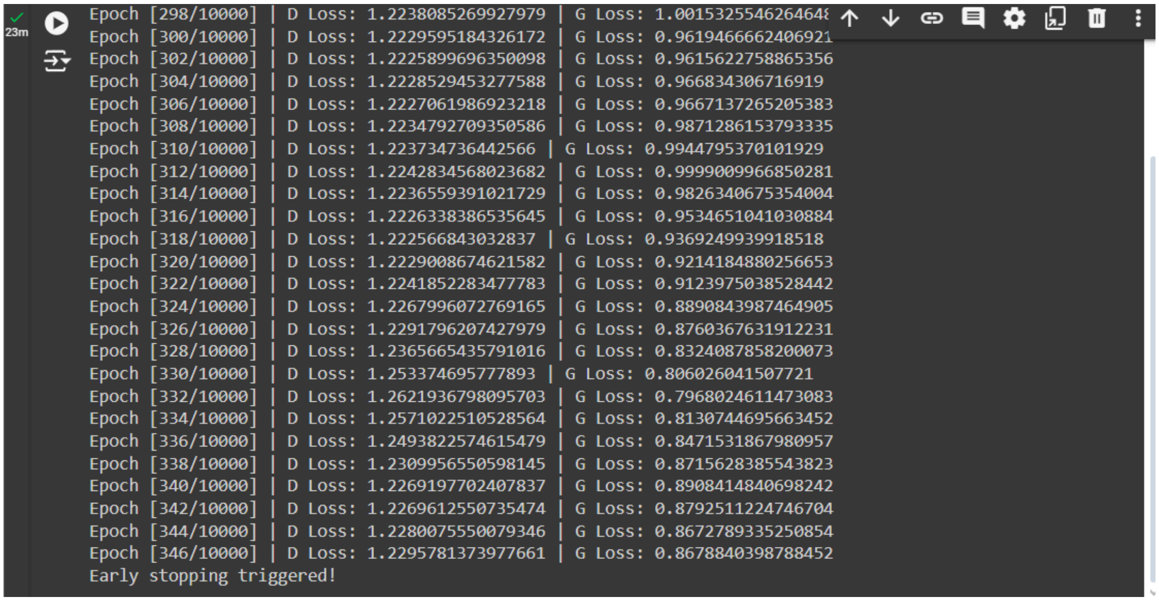


```
Epoch [264/10000] | D Loss: 1.1846728324890137 | G Loss: 1.1680716276168823
Epoch [266/10000] | D Loss: 1.2029945850372314 | G Loss: 1.2120041847229004
Epoch [268/10000] | D Loss: 1.2103469371795654 | G Loss: 1.2033565044403076
Epoch [270/10000] | D Loss: 1.197486400604248 | G Loss: 1.1746954917907715
Epoch [272/10000] | D Loss: 1.1859511137008667 | G Loss: 1.1087900400161743
Epoch [274/10000] | D Loss: 1.198326587677002 | G Loss: 1.1964821815490723
Epoch [276/10000] | D Loss: 1.1938436031341553 | G Loss: 1.1489368677139282
Epoch [278/10000] | D Loss: 1.198500394821167 | G Loss: 1.2136363983154297
Epoch [280/10000] | D Loss: 1.207581877708435 | G Loss: 1.1908913850784302
Epoch [282/10000] | D Loss: 1.1964740753173828 | G Loss: 1.194115400314331
Epoch [284/10000] | D Loss: 1.1770963668823242 | G Loss: 1.1621719598770142
Epoch [286/10000] | D Loss: 1.1922078132629395 | G Loss: 1.1358587741851807
Epoch [288/10000] | D Loss: 1.1848375797271729 | G Loss: 1.1631841659545898
Epoch [290/10000] | D Loss: 1.1934680938720703 | G Loss: 1.178707242012024
Epoch [292/10000] | D Loss: 1.1887304782867432 | G Loss: 1.1417239904403687
Epoch [294/10000] | D Loss: 1.1800113916397095 | G Loss: 1.1587661504745483
Epoch [296/10000] | D Loss: 1.191875696182251 | G Loss: 1.1138274669647217
Epoch [298/10000] | D Loss: 1.1800730228424072 | G Loss: 1.1777700185775757
Epoch [300/10000] | D Loss: 1.2013940811157227 | G Loss: 1.1479218006134033
Epoch [302/10000] | D Loss: 1.1823769807815552 | G Loss: 1.2229491472244263
Epoch [304/10000] | D Loss: 1.1932356357574463 | G Loss: 1.1358449459075928
Epoch [306/10000] | D Loss: 1.1963996887207031 | G Loss: 1.1315430402755737
Early stopping triggered!
```

Sample Generated Text: splitting dependency splitting splitting splitting lemon lemon suggested 60 60 illuminating suggested dependency dependency dependency dependency dependency dependency dependency dependency dependency dependency dependency dependency federation federation federation federation federation federation federation

federation federation federation dependency dependency dependency dependency dependency dependency dependency dependency dependency dependency dependency dependency dependency dependency remake dependency

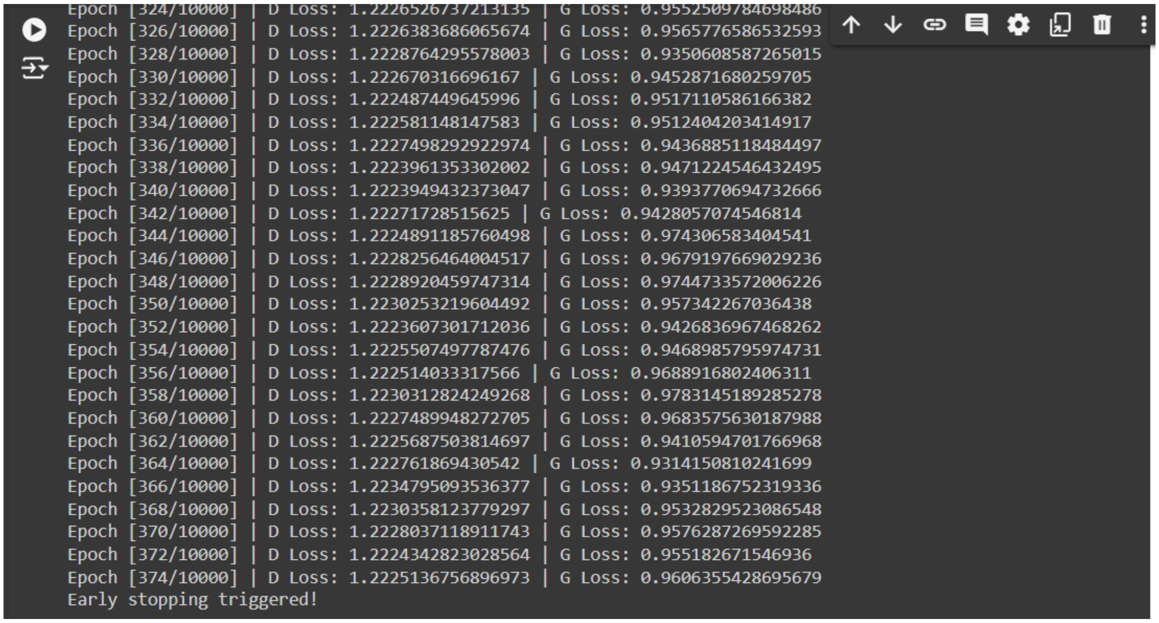
g. With Layer Normalization



```
Epoch [298/10000] | D Loss: 1.2238085269927979 | G Loss: 1.0015325546264648
Epoch [300/10000] | D Loss: 1.2229595184326172 | G Loss: 0.9619466662406921
Epoch [302/10000] | D Loss: 1.2225899696350098 | G Loss: 0.9615622758865356
Epoch [304/10000] | D Loss: 1.2228529453277588 | G Loss: 0.966834306716919
Epoch [306/10000] | D Loss: 1.2227061986923218 | G Loss: 0.9667137265205383
Epoch [308/10000] | D Loss: 1.2234792709350586 | G Loss: 0.9871286153793335
Epoch [310/10000] | D Loss: 1.223734736442566 | G Loss: 0.9944795370101929
Epoch [312/10000] | D Loss: 1.2242834568023682 | G Loss: 0.9999009966850281
Epoch [314/10000] | D Loss: 1.2236559391021729 | G Loss: 0.9826340675354004
Epoch [316/10000] | D Loss: 1.2226338386535645 | G Loss: 0.9534651041030884
Epoch [318/10000] | D Loss: 1.222566843032837 | G Loss: 0.9369249939918518
Epoch [320/10000] | D Loss: 1.2229008674621582 | G Loss: 0.9214184880256653
Epoch [322/10000] | D Loss: 1.2241852283477783 | G Loss: 0.9123975038528442
Epoch [324/10000] | D Loss: 1.2267996072769165 | G Loss: 0.8890843987464905
Epoch [326/10000] | D Loss: 1.2291796207427979 | G Loss: 0.8760367631912231
Epoch [328/10000] | D Loss: 1.2365665435791016 | G Loss: 0.8324087858200073
Epoch [330/10000] | D Loss: 1.253374695777893 | G Loss: 0.806026041507721
Epoch [332/10000] | D Loss: 1.2621936798095703 | G Loss: 0.7968024611473083
Epoch [334/10000] | D Loss: 1.2571022510528564 | G Loss: 0.8130744695663452
Epoch [336/10000] | D Loss: 1.2493822574615479 | G Loss: 0.8471531867980957
Epoch [338/10000] | D Loss: 1.2309956550598145 | G Loss: 0.8715628385543823
Epoch [340/10000] | D Loss: 1.2269197702407837 | G Loss: 0.8908414840698242
Epoch [342/10000] | D Loss: 1.2269612550735474 | G Loss: 0.8792511224746704
Epoch [344/10000] | D Loss: 1.2280075550079346 | G Loss: 0.8672789335250854
Epoch [346/10000] | D Loss: 1.2295781373977661 | G Loss: 0.8678840398788452
Early stopping triggered!
```

Sample Generated Text: ##$_e$ morally morally morally morally morally every every every morally morally morally morally morally peters peters peters morally 水 水 morally morally貴 貴 fulton fulton fulton 水 rosie rosie rosie rosie morally peters貴貴貴貴 ග ග ග ග ග peters 水 水 水 水 primitive primitive

h. With roberta-base tokenizer instead of bert-base-uncased



```
Epoch [324/10000] | D Loss: 1.2226526737213135 | G Loss: 0.9552509784698486
Epoch [326/10000] | D Loss: 1.2226383686065674 | G Loss: 0.9565776586532593
Epoch [328/10000] | D Loss: 1.2228764295578003 | G Loss: 0.9350608587265015
Epoch [330/10000] | D Loss: 1.222670316696167 | G Loss: 0.9452871680259705
Epoch [332/10000] | D Loss: 1.222487449645996 | G Loss: 0.9517110586166382
Epoch [334/10000] | D Loss: 1.222581148147583 | G Loss: 0.9512404203414917
Epoch [336/10000] | D Loss: 1.2227498292922974 | G Loss: 0.9436885118484497
Epoch [338/10000] | D Loss: 1.2223961353302002 | G Loss: 0.9471224546432495
Epoch [340/10000] | D Loss: 1.2223949432373047 | G Loss: 0.9393770694732666
Epoch [342/10000] | D Loss: 1.22271728515625 | G Loss: 0.9428057074546814
Epoch [344/10000] | D Loss: 1.2224891185760498 | G Loss: 0.974306583404541
Epoch [346/10000] | D Loss: 1.2228256464004517 | G Loss: 0.9679197669029236
Epoch [348/10000] | D Loss: 1.2228920459747314 | G Loss: 0.9744733572006226
Epoch [350/10000] | D Loss: 1.2230253219604492 | G Loss: 0.957342267036438
Epoch [352/10000] | D Loss: 1.2223607301712036 | G Loss: 0.9426836967468262
Epoch [354/10000] | D Loss: 1.2225507497787476 | G Loss: 0.9468985795974731
Epoch [356/10000] | D Loss: 1.222514033317566 | G Loss: 0.9688916802406311
Epoch [358/10000] | D Loss: 1.2230312824249268 | G Loss: 0.9783145189285278
Epoch [360/10000] | D Loss: 1.2227489948272705 | G Loss: 0.9683575630187988
Epoch [362/10000] | D Loss: 1.2225687503814697 | G Loss: 0.9410594701766968
Epoch [364/10000] | D Loss: 1.222761869430542 | G Loss: 0.9314150810241699
Epoch [366/10000] | D Loss: 1.2234795093536377 | G Loss: 0.9351186752319336
Epoch [368/10000] | D Loss: 1.2230358123779297 | G Loss: 0.9532829523086548
Epoch [370/10000] | D Loss: 1.2228037118911743 | G Loss: 0.9576287269592285
Epoch [372/10000] | D Loss: 1.2224342823028564 | G Loss: 0.955182671546936
Epoch [374/10000] | D Loss: 1.2225136756896973 | G Loss: 0.9606355428695679
Early stopping triggered!
```

Sample Generated Text: essorsessorsessorsARPARPARPARP crisesARPARP Latinos Latinos Latinos Latinos Latinos Latinos Latinos LatinosARPARP whereas whereas whereas whereas whereas whereas whereas Latinos Latinos Latinos Latinos LinkedIn LinkedIn LinkedIn LinkedInwagonwagonwagon crises crises crises crises crises crises Latinos Latinos whereas Latinos Latinos whereas