JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

SentiGEN: Synthetic Data Generator for Sentiment Analysis

Pushpika Sundarreson* and Sapna Kumarapathirage

Department of Computing, University of Westminster, London, United Kingdom;
e-mail : pushpika.20200456@iit.ac.lk; sapna.k@iit.ac.lk
* Corresponding Author: Pushpika Sundarreson

Abstract: Obtaining high-quality, diverse, accurate datasets for sentiment analysis has always been a significant challenge. Traditional approaches include annotators, which may introduce bias to datasets and are also time-consuming and expensive. These types of datasets may also not represent the variety needed to train robust and generalizable sentiment analysis models. This study introduces a novel combination of techniques to approach the problem with a novel solution. The proposed system, SentiGEN includes the use of a transformer, T5, fine-tuned and optimized using an evolutionary algorithm to generate high-quality, diverse, accurate data for sentiment analysis. The generated data is validated using XLNet to ensure high sentiment accuracy. This combination of technologies has proven successful based on the results derived from evaluating multiple models. From complex transformers such as BERT to more straightforward approaches like KNN, those trained using synthetic data demonstrated superior performance compared to their counterparts trained on real data. This enhancement in predictive accuracy was observed when evaluated on benchmark datasets such as SST-2 and Yelp. SentiGEN can generate high-quality, diverse, accurate, realistic data for sentiment analysis and successfully increased the performance of models trained on synthetic data compared to the same model trained on real data.

Keywords: Data Quality; Machine Learning; Optimization; Sentiment Analysis; Synthetic Data.

1. Introduction

Obtaining adequate, less biased, sentimentally accurate, diverse datasets has always been a challenge in sentiment analysis. Since the size and quality of the dataset play a huge role in more robust, generalizable, and accurate sentiment analysis models, it is essential to have high-quality, large datasets[1], [2]. While datasets are available for sentiment analysis models, there are circumstances in which the dataset is inadequate or incorrectly labeled, which would lead to inaccurate sentiment classification by models. Collecting and labeling data is also expensive and time-consuming. Human-labeled data are often prone to bias, and there are security and privacy concerns about using specific datasets[3]–[7]. All these issues result in limited generalization and suboptimal performance of models. Synthetic data has the potential to overcome most of these issues.

Synthetic data generation has been a long-term research process in NLP due to its significant impact and requirement in several other research fields, such as finance, the health sector, politics, etc. Accuracy, balance, size, diversity, coherence, realism, and reduced bias can result in the generation of high-quality synthetic datasets[3], [5], [8]–[11].

One of the most common issues in datasets is imbalance and the unavailability/scarcity of high-quality datasets. Although datasets are available for sentiment analysis, most of the datasets available are imbalanced, where one/two classes have more data, whereas the other classes have none or few data. This is problematic as it can cause incorrect classification of text. Most classifiers trained on imbalanced datasets tend to classify text as the majority class, which can lead to wrong conclusions and results. Imbalanced datasets can also hinder the classifier's performance. Most datasets available are targeted at social media and product reviews, leaving out other domains. These other domains have fewer or no datasets. Classifiers trained on social media/product review datasets cannot accurately classify most text from other domains, such as the health sector, due to its different topics and expressions. This is

Received: April, 9th 2024
Revised: April, 19th 2024
Accepted: April, 25th 2024
Published: April, 27th 2024

where synthetic data can be of significant use. Large synthetic datasets can be easily generated for many domains, which could also be balanced, leading to accurate classification of text[2], [5], [9], [12]–[15].

Lack of diversity and bias are other common issues faced in datasets. When imbalanced datasets tend to be biased towards the majority classes, it leads to inaccurate text classification by classifiers. Sentiment analysis datasets tend to have repetitive content in different expressions, leading to bias and less diversity. When datasets are less diverse, datasets tend to express a smaller range of emotions, which could also lead to inaccurate text classification. Generating unbiased datasets, which are also diverse, are still ongoing research areas. Synthetic datasets have the potential to overcome this problem as there is full control of the development of the dataset[1], [2], [5], [6], [9], [16], [17].

Another issue that datasets commonly have is that they may contain private data. It is unethical to use or publish another individual's private data. This issue can be avoided with synthetic datasets as artificial data presents itself as realistic. Synthetic datasets have the potential to protect individuals' privacy. However, this is still an ongoing area of research as certain techniques, such as GANs and data augmentation, may generate data that it was trained on, which may contain private data. Even though this issue persists with some techniques, a significant number of techniques have succeeded in generating synthetic data that safeguards individual confidentiality[16]–[20].

While it is important to generate data of high diversity and reduced bias, it is also necessary to ensure that it reflects realistic expressions and is consistent and understandable. This is one of the major properties that makes data high-quality. There are circumstances where data is diverse yet unclear. Unclear data can lead to incorrect classification of data by classifiers. This is a common issue faced with using GANs, as it is challenging to generate coherent data using GANs[1], [6]. Nevertheless, pre-trained models trained on extensive data have readily overcome this challenge due to their comprehensive training knowledge. Pre-trained models are currently the most successful and trending technique in carrying out several tasks in NLP. Most pre-trained models are transformer-based architectures, such as the GPT variants that have been proven successful in generating coherent, realistic data in the NLP domain[21], [22].

Most research focused on utilizing different variants of GANs and other traditional data augmentation techniques such as Back-Translation (BT) and Easy Data Augmentation (EDA) for synthetic data generation [1], [2], [5], [14], but did not consider all the aspects that make up a high-quality dataset such as balance, diversity, and sentiment accuracy.

Generating high-quality, diverse, sentimentally accurate, less biased, balanced datasets for sentiment analysis is essential as high-quality data are scarce. Training models on poor-quality datasets can hinder model performance as models will not be able to understand the complexities and nuances of sentiments. To address these challenges, a novel approach was proposed where a transformer, T5 was fine-tuned and optimized using Neural Architecture Search (NAS). The type of NAS used was an evolutionary algorithm that optimized T5 to generate high-quality, diverse, sentimentally accurate data, ensuring the generated synthetic data reflects real-world complexities and is tailored to enhance sentiment analysis models' performance. As a validation mechanism, the generated synthetic data will be validated using XLNet to ensure sentiment accuracy.

This research contributes to the field of sentiment analysis and synthetic data generation in several ways:

1. Introduces a novel combination of techniques by optimizing T5 using an evolutionary algorithm and utilizing a validation mechanism using XLNet, addressing the issue of high-quality data scarcity.
2. Generates synthetic data that accurately mirrors the complexities of real-world sentiment data, bridging the gap between synthetic and real data representations.
3. The generated synthetic data improves the performance of various sentiment analysis models, showcasing its efficacy and application.
4. Sets a new benchmark in the field by conducting comparative performance analysis of models trained on real, synthetic, and combined datasets (real and synthetic data), providing a comprehensive view of model robustness and performance.

The rest of the paper is organized as follows: a thorough literature review examines existing research. A detailed explanation of the proposed method follows this. Afterwards, the results are presented and discussed. The paper concludes with a summary of the findings,

highlighting the research's implications and suggesting potential directions for future studies that could extend the work presented here.

2. Literature Review

Although less research has been conducted on synthetic data generation for the sentiment analysis domain, the synthetic data generation domain in general, is not new and has been researched for several years. It is a rapidly evolving domain with several new advancements and approaches being found[1], [14], [19], [22], [23].

2.1. Usage of GANs to Generate Synthetic Data for Sentiment Analysis

An interesting approach was taken by [1] to generate synthetic datasets by augmentation using Sequential GAN (SeqGAN) and sentence compression using Long Short-Term Memory (LSTM) networks with attention mechanisms, as SeqGAN is unable to work well with long text. Data screening was also added to the system where a Bidirectional LSTM (Bi-LSTM) classifier would predict the sentiment of the data and remove incorrect data, resulting in a highly accurate filtered dataset. This work generated diverse data, which also increased the classification accuracy by 1%, which was marginal. The dataset bias issue was also not addressed, which reduced the data quality. Research [14] focusing on the data scarcity issue uses Conditional GAN (cGAN) to augment datasets. After cGAN was pre-trained, Gaussian random noise was injected into the data, one-sided label smoothing was used as a regularization technique, and batch normalization was also used, which successfully generated a dataset that increased the performance of classification models. However, this work only focused on the positive and negative emotions leaving out the neutral emotion which is unrealistic in a real-world scenario where all three basic categories are present. The dataset generated was imbalanced and biased towards the majority classes.

Research by [5] explored using SentiGAN and Category-Aware GAN (CatGAN) for synthetic data creation for the classes where data was scarce. This research was mainly focused on finding a solution for the class imbalance issue. It was able to successfully increase the classification accuracy and performance of a range of classifiers on the generated balanced dataset (BERT, Convolutional Neural Network (CNN), Bi-LSTM, and other classifiers) compared to the decreased performance of most classifiers on the imbalanced dataset. However, the quality and diversity of the dataset were not focused, which could have increased model performance and accuracy. A rather uncommon yet ingenious technique approached by [24] where TextGAN with an LSTM generator and CNN discriminator and TransGAN with a transformer generator and a transformer discriminator are used to generate synthetic data through augmentation of a dataset. While TextGAN showed marginal results in increasing the accuracy of the classifiers, TransGAN performed outstandingly well in increasing the accuracy of the classifiers, proving the capability of transformer-based architectures.

2.2. Usage of Data Augmentation Techniques to Generate Synthetic Data for Sentiment Analysis

Numerous researchers have utilized data augmentation methods to create synthetic data for sentiment analysis. Traditional approaches used were BT, EDA, and Word Mix-up [2], [25]–[27], Unsupervised Data Augmentation (UDA) was used [26] to augment datasets, which was successful in reducing bias in datasets, but UDA reduced the diversity of sentences. Part-of-speech (POS) focused Lexical Substitution for Data Augmentation (PLSDA) successfully increased classification model performance by augmentation of the dataset[28]. Data augmentation using Part-of-speech Wise Synonym Substitution (PWSS) and Dependency Relation-based Word Swap (DRAWS) resulted in the development of generalizable models with increased f1-scores and classification performance for aspect-based sentiment analysis[7]. In the context of Persian sentiment analysis, a novel augmentation strategy involved translating the original text into English and then back into the original language using the Google Translate API, effectively increasing the initial data volume. The position of sentences were shuffled after the completion of the translation[29]. This approach was successful in increasing the classification accuracy for Persian sentiment analysis.

While these approaches were successful in generating larger datasets and increasing the accuracy of the sentiment analysis models, since augmentation does not generate data from scratch, data quality and diversity were limited. Generating data from scratch has a higher

possibility of producing data of higher diversity and quality. The bias and imbalance issue in datasets can also be solved in this way. This approach has a higher chance of increasing the accuracy of sentiment analysis models by a greater value.

2.3. Usage of Pre-Trained Models to Generate Synthetic Data for Sentiment Analysis

GPT-3 was used to label the sentiment for a collected set of tweets. The labeled dataset was used for Thai sentiment analysis, which successfully increased the classification models' accuracy[22]. However, GPT-3 was only used for labeling instead of generating the whole dataset, resulting in limited data diversity, novelty, and quality. This capability could have been utilized since GPT-3 can generate high-quality data from scratch. In another research, Pre-trained Data AugmenTOR (PREDATOR) and BART were used to augment datasets by generating new sentences or parts of the existing sentences in the dataset according to the pattern of the dataset[2]. PREDATOR performed exceptionally well compared to BART. PREDATOR increased the classification accuracy and performance, whereas results from BART were marginal. Even though model performance for sentiment analysis was increased, data quality was not focused upon. As data generated through data augmentation generates data according to the specific pattern of the dataset, data diversity was also limited. High-quality data is key to model generalization, performance, accuracy and robustness.

2.4. Other Synthetic Data Generation Techniques for Sentiment Analysis

k-means and Density-based Spatial Clustering of Applications with Noise (DBSCAN) were used to break down and cluster classes to understand data trends. Class Decomposition Synthetic Minority Class Oversampling (CDSMOTE) was used to generate synthetic datasets for sentiment analysis after the breakdown of classes [9]. Although CDSMOTE-DBSCAN outperformed CDSMOTE-k-means in producing balanced, unbiased datasets and increasing the performance of classification models, data quality was not focused upon.

2.5. Review of Technologies Utilized in Synthetic Data Generation

2.5.1. Generative Models

a. Introduction to GANs

Introduced in 2014, GANs feature a dual-component system: a generator and a discriminator. The generator's role is to produce data, and the discriminator evaluates whether this data is real or fabricated. This process iterates continuously until the generator can produce data indistinguishable from actual data. Unlike the discriminator, which has access to both real and synthetic data, the generator learns to produce data solely based on feedback from the discriminator. Both the generator and discriminator are types of neural networks working together to refine the quality of the synthetic data produced. GANs are used mostly in unsupervised learning. GANs are usually computationally intensive due to their structure and nature. The continuous requirement for generating data using the generator while the discriminator continuously pushes the generator to generate realistic data is time-consuming and resource-intensive. GANs exhibit instability due to the competitive min-max optimization dynamic between the generator and discriminator[8], [11], [20], [30]–[32].

b. Architecture of GANs

cGAN: cGAN stands for a specialized form of GAN, which shares the core components of a generator and a discriminator. What sets cGAN apart is its use of conditions to steer the generation and evaluation processes. In this setup, both the generator and discriminator are structured as feed-forward neural networks operating under specific conditions. This arrangement allows the generator to create text tailored to these guidelines. At the same time, the discriminator critiques the generator towards producing outputs that better align with the given conditions, ultimately enhancing the precision and relevance of the generated content[14].

SeqGAN: The generator in SeqGAN is an LSTM whereas a Convolutional Neural Network (CNN) is the discriminator. This combination is typically used for numerical data. Given the complexity of text compared to numerical data, SeqGAN incorporates Reinforcement Learning (RL), where the generator is rewarded or penalized based on the grammatical accuracy and variety of its generated text. This approach allows the generator to refine its output

by learning from the discriminator's feedback and insights gained through RL, enhancing the quality of the text it produces[1].

CatGAN: CatGAN consists of the category-aware model and the hierarchical evolutionary algorithm. The generator contains a Relational Memory Core, which compares the fabricated text and the original text, attempting to reduce the distinction between the original text and the fabricated text. The hierarchical evolutionary algorithm is the training strategy that also acts as the discriminator, where it differentiates the fabricated text from the original text and provides feedback so that the generator improves the category text generation[5], [33].

SentiGAN: SentiGAN is comprised of several LSTM-based generators and a single classifier. LSTM generators work independently and are not influenced by each other. The generator uses the Monte Carlo Search to explore different writing styles and fine-tunes itself to use the best-suited writing style. The classifier classifies the text and checks if the correct sentiment is generated and if the text generated is realistic. Feedback from the classifier helps the generator improve the text generated. SentiGAN also consists of a penalty-based system which ensures that the generated text contains the correct sentiment[5].

TextGAN: TextGAN consists of an LSTM generator and a CNN discriminator. The generator generates sentences sequentially by predicting the next word. The discriminator checks if the generated text is synthetic or real and provides feedback to the generator, helping improve the text generated. The generator aims to produce text indistinguishable from the original text, mirroring the function of a standard GAN[24].

TransGAN: TransGAN functions similarly to a typical GAN. The only difference between a GAN and a TransGAN is that the generator and the discriminator are transformers [24].

Synthetic Data Generation GAN (SDG-GAN): Both the generator and the discriminator operate as feed-forward neural networks, utilizing a Multilayer Perceptron (MLP) design. The generator is tasked with creating data that closely resembles the characteristics of genuine data, leveraging features learned from actual data during its training. Meanwhile, the discriminator evaluates whether the data is fabricated or authentic by examining it against features derived from real data. SDG-GAN is specially designed for imbalanced datasets[16].

CTGAN: CTGAN features a generator and a discriminator, both of which are types of neural networks. The generator generates data based on a given condition, resulting in more diverse data, whereas the discriminator uses PacGAN, which differentiates real and fake data in groups. This is proven to be better at differentiating real and fake data[23], [32], [34].

UniformGAN: Neural networks are utilized for the generator. The generator also uses a Scaled Exponential Linear Unit activation, which helps improve training. The generator adds noise to the data to generate better realistic synthetic data. It utilizes a technique where the generator is penalized if the data generated is not equal across a range. This can result in reduced bias in data. The discriminator is a CNN using Scaled Exponential Linear Unit activation[19].

Medical Text GAN (mtGAN): mtGAN generates electronic medical records based on a condition. Like SeqGAN, mtGAN also uses a reinforcement learning technique called REINFORCE, a policy gradient algorithm to generate data that allows for the realistic generation of electronic medical records. The data generated successfully protected the privacy of real data while maintaining diversity. However, this technique had less control over the data generated[13].

c. Other Generative Models

Categorical Latent Gaussian Process (CLGP): CLGP takes complex data as input and presents them in a simplified version on a latent space. The Gaussian Process grasps the patterns between the data. The Softmax function identifies the attributes of the data and assigns a value to each feature, enabling the creation of synthetic data that closely replicates the original data trends. This process increases the computational workload due to the Gaussian Process requiring a lot of time to do calculations, especially for complex, large data[8].

2.5.2. Data Augmentation

a. Introduction to Data Augmentation

This is a technique that generates artificial data using the existing data by making modifications to it and presenting the modified data as new data. This approach increases the size of the existing dataset without gathering new data, yielding both the original and the altered data as outcomes. While data augmentation is a convenient way of generating data using

existing data, diversity is limited as data is not generated from scratch. Bias may also be introduced into the data unknowingly, as there is less control over how the data is generated[2], [25], [27], [35].

b. Architecture of Data Augmentation Techniques

BT: This augmentation method involves translating the text from its original language to another chosen language and then back to the original language. The result is the text of a similar sense but presented in a different form. However, the effectiveness of this technique depends on the languages chosen as it may not work for all languages[2], [25], [27], [35], [36].

Word Mix-up: This technique arranges sentences in a way that allows all sentences to be of equal length. The sentences are shuffled while maintaining coherence and meaningfulness. However, it can be challenging to maintain semantics in the shuffling process[25], [36].

EDA: This technique involves altering words while maintaining semantics, introducing alternate words at different positions of sentences, interchanging words, and removing words. EDA can introduce repetitiveness into the data, can be challenging to maintain semantics during the augmentation process, and can lose important information embedded in sentences during the removal process[2], [25], [27], [35], [36].

PLSDA: PLSDA uses POS tags to guide the replacement of synonyms, with a 50% chance of the synonym being swapped. The word being replaced is selected based on grammatical accuracy and semantics[28].

PWSS: POS tags and Spacy are used to identify words suitable to be swapped with and replaced with words from WordNet, a repository of English words based on grammatical correctness[7].

DRAWS: The structure of sentences is observed, and words are identified based on the connection to the main topic structure, substituting the identified words between similar sentences. Words are also modified or removed if needed. The entire process depends on the relationship to the main topic[7].

Contextual Word Embeddings (CWE): In this technique, words are identified and replaced by using Large Language Models (LLM) such as BERT[35].

2.5.3. Pre-trained Models

a. Introduction to Pre-trained Models

Pre-trained models, often deep learning models, are developed using extensive datasets. They possess a wide range of knowledge across numerous domains and can be adapted or fine-tuned for various specific tasks, including text classification and synthetic data creation. Due to their vast knowledge, these models are more accurate and robust, have better performance, generalization, and are more computationally intensive than other models due to the immense amount of data it was trained on, hyperparameters, and layers they possess[21], [37].

b. Architecture of Pre-trained Models

BART: This auto-regressive transformer mainly rephrases text, which can be a useful technique when attempting to generate synthetic data or augment data. The encoder derives information from the input. The decoder uses the masked multi-head attention mechanism and the Beginning of the Statement (BOS) token. This token specifies the topic to generate text initially, ensuring pertinence, fluency, and grammatical correctness[2], [35], [38].

PREDATOR: PREDATOR uses pre-trained models for its generator and filter. The generator creates new text while the filter removes the low-quality text and only outputs the final data containing high-quality data[2], [39].

GPT Variants: GPT is an auto-regressive transformer with several versions – GPT-2, GPT-3, GPT-3.5, GPT-3.5 Turbo and GPT-4. While GPT-3.5 Turbo and GPT-4 are relatively new and have not been researched widely, the other previous versions have been used to generate text. While other differences remain, one main difference between all models is the amount of data they were trained on and the number of hyperparameters they have, making the latest versions more powerful and precise due to increased knowledge and hyperparameters[35], [40]. GPT utilizes a multi-head self-attention mechanism to process input, uses layer normalization to enhance training stability, and adopts Byte Pair Encoding (BPE) to tokenize the input. Meanwhile, its decoder produces text by leveraging these methods[41]–[43].

2.5.4. Reinforcement Learning

a. Introduction to Reinforcement Learning

RL involves an agent that engages with the environment and receives continuous feedback through penalties and rewards. This feedback encourages the agent to continuously explore new actions in the environment to maximize the rewards, resulting in the agent excelling at its function with time[44], [45].

b. Architecture of Reinforcement Learning Techniques

Reinforcement Learning-based Text Generator (RLTG): RLTG utilizes a language model for word recommendation, utilizing Deep Q-learning, a form of RL technique, as its RL agent to create text based on these suggestions. An adversarial RL mechanism assesses the realism of the generated text, offering feedback that translates into rewards or penalties for the agent. This process progressively enhances the agent's ability to produce realistic data, optimizing for maximum rewards[44].

2.5.5. Neural Networks

Recurrent Neural Networks (RNN) and LSTM: RNN represents a category of neural networks that can understand trends in data. The loops in RNN can pass information from one level to another in the networks, allowing it to recall the previous data in the loop. However, it cannot retain information in the long term. This is where LSTM, an RNN, excels at retaining information long-term. This is an advanced version of RNN with a built-in mechanism that allows it to update, recall, or forget information selectively. RNN and LSTM generate realistic synthetic data by remembering the patterns from the training data[12], [46].

Autoencoders (AE) and Variational Autoencoders (VAE): AE comprises an encoder and a decoder. The encoder compresses the incoming data into a latent space representation at a fixed point. The decoder regenerates the data in a different form. VAE is a type of AE that is similar to AE, but the encoder in VAE presents the data in a latent space as a range instead, which allows for better generalization and pertinent data[23], [32], [47].

Differentially Private Synthetic Data Generation (DP-SYN): This method divides the data into clusters, utilizing a distinct AE for each cluster to identify and learn the trends unique to that group. It then creates synthetic data for each cluster, reflecting the trends observed [18].

2.5.6. Probabilistic and Stochastic Techniques

Copula: Copulas alone cannot generate synthetic data. In novel research studies of Copulas, Copulas takes uniform random variables and converts them so all are related. Correlation is introduced between these variables. By applying inverse transform sampling on the correlated variables, new data is generated that represents the same patterns in the original data [17], [48], [49].

Bayesian Networks (BN): Using graphs, BN shows the relationships between variables. Synthetic data is generated by observing the relationships in real data using the Chow-Liu tree. Even though it is efficient, it can miss out on important information if unable to observe all relationships between variables, making it most suitable for discrete data[8], [23]

Markov Chains: Markov Chains create a chain where each state represents a word from the chain. The initial word is chosen based on the frequency of it in the text or a user-defined condition. The following words are predicted based on the patterns learned in a given data and the preceding words in the data, successfully generating text that imitates the pattern and style of the original data[12].

Multivariate Imputation by Chained Equations (MICE): First, a sequence for the variables is created. The initial variable is selected based on empirical distribution. Subsequent variables are forecasted with a probabilistic model that estimates the next word by considering the preceding variable in the sequence. These variables form the synthetic data[8].

Independent Marginals (IM): This is a more straightforward technique that generates synthetic data by observing each variable separately without considering the relationships between other variables (empirical marginal distribution), resulting in less realistic data as it is unable to consider complex relationships[8].

Mixture of Product of Multinomials: Probability models are created for each pattern identified in the data. Weights are allocated to models based on how frequently the pattern is identified. The Dirichlet process is used to identify the different patterns to utilize in the

model. Gibbs sampling, a Markov Chain Monte Carlo algorithm variant, generates new data based on the identified patterns in the original data[8].

Gaussian Mixture Model (GMM): GMM assumes that all data can be represented as Gaussian distributions. Each Gaussian has a mean and standard deviation. The number of Gaussians is pre-decided. The expectation-maximization algorithm is utilized to fine-tune Gaussians, ensuring that all data fits well. After the finalization of the parameters, random sampling of the Gaussian distributions can generate synthetic data[23].

2.5.7. Sampling Techniques

Random Oversampling: This method does not create new data. It merely duplicates the existing underrepresented class data to equal the size of the dominant class, leading to less diversity and overfitting. This technique is less preferred for data generation as it only replicates data but can help with imbalanced classes in data[23].

Synthetic Minority Oversampling Technique (SMOTE) Variants: SMOTE causes interpolation between the minority classes. It selects a data point belonging to the less represented class and another close data point to create new data along that sequence of data points. This process is repeated until the necessary amount of data is generated. This technique tends to be biased as it focuses on the minority class, ignoring the majority class and reducing diversity as well. Borderline-SMOTE and Safe-level-SMOTE are different improved versions of SMOTE. Borderline-SMOTE focuses on generating samples that could be misclassified in the minority class, whereas Safe-level-SMOTE focuses on generating data that are concentrated with data points from the minority class within the attribute space. Adaptive Synthetic Sampling Approach for Imbalanced Learning (ADASYN) and SMOTE are similar. ADASYN generates data for the minority class data that classifiers find difficult to classify. The density distribution does this. This technique helps improve classifier performance as more samples that classifiers find challenging to classify will be available for the classifier to learn from. This leads to reduced bias and prevention of overfitting. K-means-SMOTE is another variant of SMOTE where K-means clustering is applied to the data to identify and form data clusters. Some clusters of the minority class are removed. SMOTE was applied to the remaining clusters afterward. This technique allows for better generalization and less noise as it focuses more on the data clusters, increasing data quality. Cluster-based Oversampling forms data clusters of all classes, which organize the data, and random oversampling is applied afterward. All of the SMOTE variants are similar and focus on providing a solution for data class imbalance. With the evolution of technologies, these techniques are less preferred over new technologies like GANs and pre-trained models due to their advancements. SMOTE variants are also computationally intensive for large data and cannot capture the complex structures in data[23].

2.5.8. Other Synthetic Data Generation Techniques

Faker: Faker is an open-source engine that can generate realistic synthetic data, including mobile numbers, account details, addresses, zip codes, prices, product names, and more[17].

Universal Text Generator (UTG): UTG is a library that can generate text according to specific rules and conditions like variables and characteristics. It also maintains coherence and grammar[40].

2.6. Evaluation Methods

The evaluation of synthetic data generated for sentiment analysis consists of two parts:

1. Evaluating the quality of the generated dataset using a variety of evaluation metrics[1], [14].
2. Testing out the generated dataset by training classifiers on the generated dataset[1], [14].

2.6.1. Evaluation Methods for the Dataset

1. Jaccard Similarity: This technique measures the diversity of the data by comparing each data point to another data point in the data but does not compare itself to provide a similarity score. The lower the similarity, the higher the diversity[1].
2. Negative Log-Likelihood (NLL): NLL checks how probable the data is from the model that generated the data. A low NLL value means better performance, better generalizability, and diversity. However, a too-low NLL value means overfitting, so a moderate NLL value is recommended[5], [33].

3. Bilingual Evaluation Understudy (BLEU): This metric evaluates the n-grams between the reference text and the text produced, imposing penalties on shorter texts to determine a final score. A high score means the text is very similar to the reference text, which has low diversity and novelty. A low score means high diversity and high novelty [5].
4. Recall-Oriented Understudy of Gisting Evaluation (ROUGE): Like BLEU, this measure assesses the resemblance of the produced text to the reference material. A high rating indicates a close similarity to the reference, signaling low diversity and novelty. A low score means high diversity and high novelty[26].
5. t-distributed Stochastic Neighbor Embedding (t-SNE): This is a representation method where a set of high-dimensional data is presented on a 2D or 3D map. This helps to understand the patterns and clusters in the data, which were initially difficult to understand in the high-dimensional data. Real data and fabricated text can be compared and visualized using t-SNE. The feature space that the authentic data occupies and the feature space that the fabricated text occupies can be compared on one flat space. This helps to understand if the synthetic data covers all the features and clusters the real-world data covers[14], [50].

2.6.2. Evaluation Methods for the Classifier

1. Accuracy: The classification accuracy of the classifier is evaluated to check how well the model classifies sentiments on the synthetic dataset[1], [5], [14], [22], [24], [27]–[29], [51].
2. Precision: Precision quantifies the proportion of accurately predicted instances among those identified by the model as correct[9], [27], [51].
3. Recall: Recall assesses the proportion of correctly identified instances by the model among all actual occurrences that were correct[9], [27], [51].
4. F1-Score: The f1-score combines both precision and recall, providing a balanced measure of performance[2], [5], [7], [9], [27], [51].

Based on the literature review conducted, higher-quality datasets for sentiment analysis are required that mirror the complexities of real-world data. Most researchers used GANs or different data augmentation techniques to increase the size and quality of the dataset produced. However, these techniques did not look into all the properties that make a high-quality dataset, such as balance, diversity, bias, accuracy, and size. While some of these properties were taken into account, all of the properties were not satisfied, which reduced the data quality [1], [14], [15], [25].

Based on current research, deep learning models and transformer-based architectures have successfully generated text-based data[52]. According to the current literature, although T5, a transformer-based deep learning model, has been used for generating text, it has not been used widely to create synthetic datasets for sentiment analysis[52]–[55].

Based on the literature review, NAS has not been utilized to optimize transformer models within the synthetic data generation and sentiment analysis domain to generate high-quality, accurate, realistic, and diverse data for sentiment analysis. This presents a new research opportunity that can be explored by optimizing T5 using NAS to generate high-quality, accurate, and diverse synthetic sentiment data. Fine-tuning and optimizing T5 using NAS and XLNet to validate the NAS-optimized-T5-generated data for producing high-quality, accurate, and diverse synthetic sentiment data remains an unexplored area yet to be explored. Successfully implementing these techniques can enhance dataset quality, resulting in improved performance, robustness, generalization, and accuracy of sentiment analysis models.

3. Proposed Method

This research introduces a novel approach to enhance synthetic data generation for sentiment analysis through the optimization of T5 using an evolutionary algorithm - Distributed Evolutionary Algorithms in Python (DEAP) [56] framework and leverages two key diversity metrics, Jaccard Similarity and Self-BLEU scores, to evaluate the diversity and quality of the generated data. The DEAP framework facilitates an evolutionary approach to optimize model parameters systematically, enabling the generation of high-quality, diverse synthetic data. Afterward, XLNet classifies the sentiments generated by the optimized T5 to ensure the sentiment accuracy of the generated text.

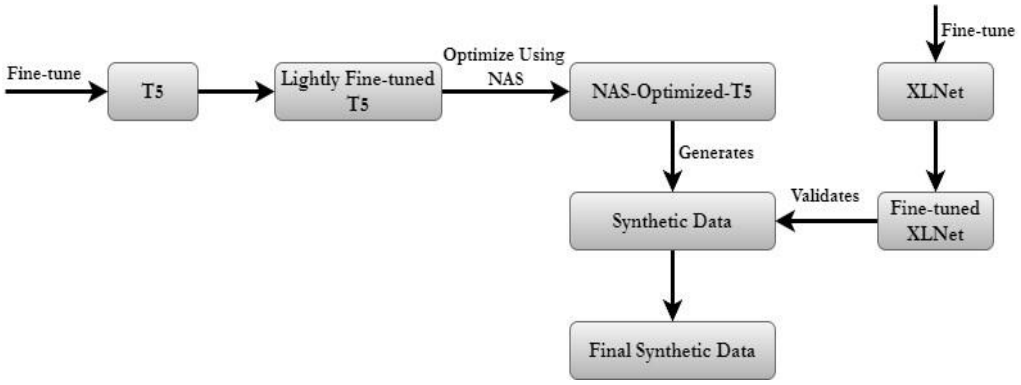

Figure 1. The general flow of the proposed method

Firstly, T5 is lightly fine-tuned over just two epochs to give the model preliminary knowledge of the task of generating text for sentiment analysis. The fine-tuned T5 is optimized using DEAP to enhance the fine-tuned model further. The DEAP framework provides the platform for the evolutionary optimization process, enabling the efficient search and selection of optimal hyperparameters for the T5 model. Through genetic algorithms, a search space, including learning rate, batch size, epochs, dropout rate, weight decay, and a custom loss weight aimed at encouraging data diversity, which determines the weight/impact of the custom diversity-encouraging loss function on the model training which is incorporated into the training function. Each set of parameters, or individual in the evolutionary context, undergoes mutation and crossover operations to produce new parameter sets, with selection based on fitness scores derived from the diversity evaluation metrics.

This diversity-encouraging loss function penalizes the generation of repetitive or similar sentences while promoting semantic coherence. It combines a token diversity penalty, which discourages repetitive token use, with a semantic coherence loss, minimizing the cosine similarity between successive sentence embeddings generated by the model. This encourages the model to explore fewer common tokens and produce diverse, contextually relevant sentences.

Jaccard Similarity and Self-BLEU metrics are utilized to assess the effectiveness of the optimized T5 model. The Jaccard Similarity measures the uniqueness of the generated sentences by evaluating the overlap between the sets of tokens in different sentences. Lower Jaccard Similarity indicates higher textual diversity. The Self-BLEU score assesses the degree of similarity between a sentence and the other sentences generated by the model, with lower scores indicating greater diversity. The evolutionary algorithm's fitness function identifies optimal parameters by aiming for models that generate diverse text, as indicated by the low Jaccard Similarity and Self-BLEU scores.

Table 1. Evolutionary algorithm settings

Setting	Selection
Population Size	5
Number of Generations	6
Crossover Rate	0.8
Mutation Rate	0.1
Selection Strategy	NSGA-II
Evaluation Metrics for Model Selection	Self-BLEU, Jaccard Similarity

The model's training procedure incorporates the AdamW optimizer. This OneCycleLR learning rate scheduler starts with a lower learning rate that gradually increases to the maximum before annealing and selective weight decay, where the weight decay is not applied to biases and normalization layer weights. Gradient accumulation is also incorporated to manage the computational demands of larger batch sizes and stabilize the training. Once the evolutionary algorithm finds the optimal parameters and the optimum diversity-encouraging loss function weight, the final T5 model is trained. The optimized T5 is used to generate synthetic data for sentiment analysis. Each piece of generated data undergoes a sentiment classification check by XLNet to confirm its sentiment accuracy. Only the data where XLNet's

classification aligns with the sentiment labels initially generated by T5 are incorporated into the final dataset, ensuring both accuracy and consistency in labeling. Table 1 shows the evolutionary algorithm settings, and Table 2 shows the selected search space settings.

Table 2. Search space settings

Setting	Selection Range
Learning Rate	1e-5 – 4e-5
Batch Size	5 - 9
Epochs	1 - 7
Dropout Rate	0 – 0.5
Weight Decay	0 – 0.05
Diversity-Encouraging Loss Weights	0.01 – 0.5

4. Results and Discussion

The validation model, XLNet, achieved an accuracy of 83.6% and an f1-score of 0.836. The parameters utilized to fine-tune XLNet were as follows:

Table 3. XLNet fine-tuning parameters

Hyperparameter	Selection
Learning Rate	3e-5
Batch Size	46
Epochs	2
Weight Decay	0.01

The generated synthetic dataset was evaluated using a variety of evaluation metrics, including Jaccard Similarity, ROUGE, and BLEU, to calculate the diversity of the text generated. t-SNE analysis was also performed to visualize the real and synthetic data on a 2d platform. The implementation and testing of this study were carried out using the A100 GPU on Google Colab, leveraging its powerful computing capabilities.

4.1. Jaccard Similarity, ROUGE and BLEU

Table 4. Jaccard Similarity and BLEU scores

Evaluation Metric	Score
Jaccard Similarity	0.112
BLEU	0.0029

Table 5. ROUGE scores

Evaluation Metric	Recall	Precision	F1-Score
ROUGE-1	0.148	0.191	0.123
ROUGE-2	0.008	0.015	0.008
ROUGE-L	0.134	0.173	0.110

In assessing dataset diversity, the Jaccard Similarity measures the overlap between pairs of data samples. It is calculated by dividing the count of shared attributes by the count of all distinct attributes across both samples. Diversity is then determined by subtracting the maximum Jaccard Similarity found among all unique pairs from one. Lower scores of Jaccard Similarity indicate greater diversity, pointing to a broader array of distinct elements within the dataset. Jaccard Similarity of two data points, I_1 and I_2 are expressed in Equation (1)[1].

$$\mathrm{Sim}(I_1, I_2) = \frac{|\, I_1 \cap I_2 \,|}{|\, I_1 \cup I_2 \,|} \tag{1}$$

The inverse of the highest resemblance is viewed as the diversity of the data and is expressed in Equation (2)[1].

$$\text{Diversity}(I_s) = 1 - \max\{\text{Sim}(I_s, I_t)\}\, s, t \in \{1,2,\cdots n\}, t \neq s \tag{2}$$

A lower Jaccard Similarity score indicates a higher diversity. This indicates that the generated dataset is diverse. A score of 0.112 is closer to 0, indicating high diversity in the data generated.

Brevity penalty (BP) and BLEU are expressed in Equation (3) and (4)[57].

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{\left(1-\frac{r}{c}\right)} & \text{if } c \leq r \end{cases} \tag{3}$$

$$\text{BLEU} = BP \times exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{4}$$

Equation (4) incorporates a BP to account for the length of the generated text compared to the reference text. There is no penalty if the generated text is longer than the reference. The BLEU score itself is the product of the BP and the exponentiated average log precision of matched n-grams between the generated and reference texts, weighted by w_n. N-gram matches of the generated text against the reference text are evaluated to determine their similarity. For diversity assessment, lower n-gram precision would yield a lower BLEU score, signaling that the generated text varies from the reference and thus contains a richer variety of language use. In short, the brevity penalty ensures that the generated text is not trivially short, while the weighted precision captures the novelty of the generated text.

Table 4 shows the BLEU scores of the generated data. Lower BLEU scores indicate that the generated text differs from the reference text (training data). Since the BLEU score is closer to 0, this indicates that the model was successful in generating data that was significantly different from the training data, successfully generating diverse data.

ROUGE-1 measures the overlap of unigrams (single words). ROUGE-2 measures the overlap of bigrams (pairs of consecutive words). ROUGE-L measures the longest common subsequence. Table 5 shows the ROUGE scores of the generated data. Lower ROUGE scores indicate that the generated text differs from the reference text (training data). Since the ROUGE scores are closer to 0, this indicates that the model successfully generated data that was significantly different from the training data, successfully generating diverse data.

4.4. t-SNE Visualization

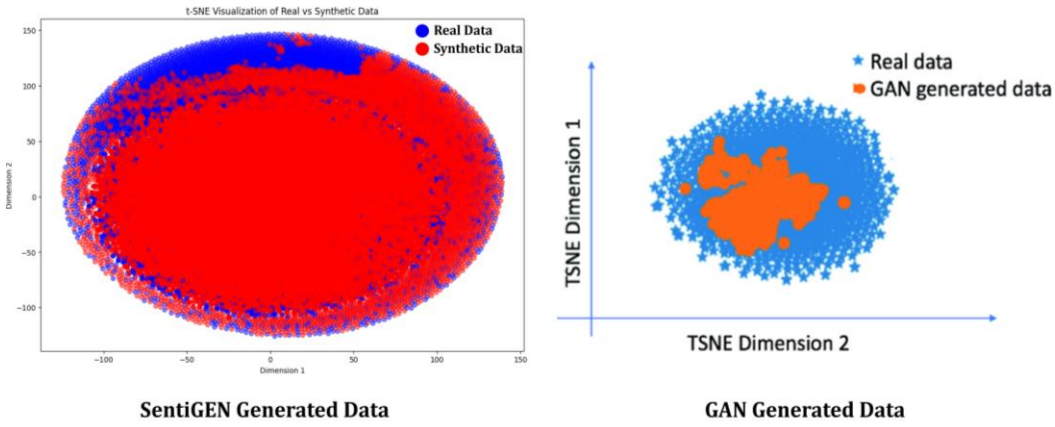

Figure 2. t-SNE visualization comparison between SentiGEN and GAN-generated data

Figure 2 shows a t-SNE visualization of real data and synthetic data generated by a GAN in research [14] on the right side whereas the left side of Figure 2 shows a t-SNE visualization of real data and synthetic data generated by SentiGEN. SentiGEN generates data very similar to real data and covers almost the entire feature space that real data covers. According to the t-SNE, SentiGEN surpasses the GAN-based model in generating diverse data that mirrors the complexities and nuances of real-world data.

4.5. Comparative Performance Analysis: Models Trained on Real, Synthetic, and a Combination of Real and Synthetic Data

A range of classifiers were trained on the synthetic data, real data, and a dataset, which consisted of an equal number of real data and synthetic data for model performance comparison.

1. Real data – 60,000 samples
2. Synthetic data – 60,000 samples
3. Real data + synthetic data – 30,000 samples of real data and 30,000 samples of synthetic data.

The training conditions remained the same across the three datasets (real, synthetic, and combined) for each model to make a fair comparison and to check how each dataset type affects the model's performance. All the trained models (real, synthetic, and combined) were evaluated on the SST-2 benchmark dataset and the Yelp benchmark test dataset, which was not a part of the training dataset and, as a result, was completely unseen to all the models. The SST-2 dataset consists of positive and negative sentiments. For the model evaluation using the SST-2 dataset, 15,000 positive samples and 15,000 negative samples were utilized. The Yelp dataset consists of positive, neutral, and negative sentiments. For the model evaluation using the Yelp dataset, 5000 positive samples, 5000 neutral samples, and 5000 negative samples were utilized.

Table 6 and Table 7 provide a comparative analysis of the performance of various models across two different datasets: SST-2 and Yelp. Each table showcases the models' performance when trained on three data types: real, synthetic, and a combination of real and synthetic. The tables clearly indicate the impact of training data type on model accuracy and F1-score, which are critical measures of a model's predictive capabilities and the balance between precision and recall, respectively.

When examining models trained on real data, a general trend can be observed in both datasets where transformer-based models demonstrate high accuracy and F1-scores, indicating their robustness in understanding complex language patterns found in real-world data. On the other hand, traditional machine learning models such as Decision Trees show significantly lower scores, suggesting a potential inadequacy in capturing the intricacies of natural language compared to deep learning models.

Table 6. Performance comparison of multiple models on SST-2.

Model	Real Data		Proposed Method (Synthetic Data)		Real Data + Proposed Method (Synthetic Data)	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
XLNet	84.01	0.858	88.43	0.890	85.69	0.872
BERT	82.61	0.846	86.92	0.872	84.18	0.856
RoBERTa	84.68	0.867	88.03	0.884	86.46	0.876
DistilBERT	81.24	0.833	85.14	0.856	82.63	0.842
ERNIE	84.42	0.858	88.19	0.885	84.72	0.862
Decision Trees	47.21	0.520	54.10	0.560	48.53	0.530
Random Forest	54.94	0.570	61.35	0.600	61.63	0.600
Logistic Regression	66.94	0.710	68.09	0.690	67.08	0.700
KNN	57.76	0.610	62.62	0.640	60.35	0.630
SVM	66.85	0.710	69.91	0.700	67.56	0.710
Adaptive Boosting	51.77	0.570	58.46	0.590	52.54	0.570

Table 6 shows that models trained on synthetic data and those trained on a combination of real and synthetic data outperform those trained on real data.

On both datasets, all models trained on synthetic data and models trained on the hybrid dataset (synthetic + real data) show superior performance compared to their counterparts trained on real data, indicating the synthetic data's quality, representativeness, and accuracy. On the SST-2 dataset, models trained on synthetic data perform best. However, while some models excel with synthetic data on the Yelp dataset, some models, like DistilBERT achieve better results on the hybrid dataset. The varying performances also highlight how the source

of the training data can impact model effectiveness. The contribution of synthetic data to improved model performance emphasizes the optimization of the T5 model and the high fidelity of the synthetic data it generates. Overall, the tables reveal that synthetic data positively impacts the model's ability to generalize and accurately predict sentiments. These findings show that synthetic data can be crucial in addressing data scarcity and diversity issues. Additionally, the results show the value of exploring hybrid datasets as a means to potentially leverage the strengths of both real and synthetic data for improved model performance.

Table 7. Performance comparison of multiple models on Yelp.

Model	Real Data		Proposed Method (Synthetic Data)		Real Data + Proposed Method (Synthetic Data)	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
XLNet	78.43	0.788	79.88	0.802	79.83	0.801
BERT	77.35	0.778	78.62	0.791	78.41	0.788
RoBERTa	79.23	0.795	79.62	0.800	80.03	0.804
DistilBERT	76.41	0.768	77.52	0.780	77.79	0.782
ERNIE	75.25	0.759	79.17	0.796	79.23	0.796
Decision Trees	50.21	0.500	51.48	0.520	50.91	0.510
Random Forest	57.05	0.570	61.15	0.610	61.04	0.610
Logistic Regression	69.52	0.700	69.80	0.700	70.28	0.710
KNN	46.46	0.450	51.73	0.520	49.71	0.490
SVM	69.14	0.700	69.95	0.700	69.97	0.710
Adaptive Boosting	62.91	0.640	64.33	0.650	65.13	0.660

Table 7 shows that models trained on synthetic data and those trained on a combination of real and synthetic data outperform those trained on real data.

4.6. Comparative Analysis: Model Performance on NAS-Optimized-T5-Generated Synthetic Data with and without Custom-Diversity Encouraging Loss Function

Table 8 compares model performance, specifically accuracy scores, using two types of training data sources: synthetic data generated by NAS-optimized T5 models, both with and without a custom diversity-encouraging loss function. The accuracy scores show that incorporating the custom diversity-encouraging loss function during the optimization of T5 using NAS has a positive impact across all models. Models utilizing synthetic data generated by the T5, which included the diversity-encouraging loss function in its optimization, demonstrated superior performance compared to those trained on data produced without this specialized loss function. This indicates the effectiveness of the custom loss function in improving the quality of synthetic data for training sentiment analysis models, as reflected in the enhanced accuracy scores across the table.

Table 8. Models' accuracy comparison with and without the custom diversity-encouraging loss function on SST-2

Model	Training Data Source	
	NAS-Optimized T5 Without Custom Diversity-Encouraging Loss Function	NAS-Optimized T5 With Custom Diversity-Encouraging Loss Function
XLNet	87.85	88.43
BERT	86.31	86.92
RoBERTa	87.73	88.03
DistilBERT	84.75	85.14
ERNIE	87.66	88.19

Table 8 shows that all models trained on synthetic data generated by the NAS-optimized T5 with the custom diversity-encouraging loss function have increased accuracy.

5. Conclusions

The primary goal to refine and enhance the T5 transformer through NAS, supplemented by XLNet's validation of text outputs, was to create high-quality synthetic text that accurately reflects diverse sentiments. This objective has been effectively achieved, as evidenced by various performance indicators. The application of metrics such as BLEU, ROUGE, Jaccard Similarity, and t-SNE visualization collectively attest to the synthetic data's quality, diversity, and fidelity. Furthermore, models trained on this synthetic dataset demonstrated superior performance compared to those trained solely on real data, reinforcing the synthetic data's effectiveness and the successful realization of the project's aim. The generated data also mirrored the complexities and nuances of real-world data. This research also sets a new benchmark by comparing the performance of models trained on real and synthetic data and a combination of both real and synthetic data. This unique combination of techniques has the potential to open up new research avenues, where this combination of techniques can be applied to generate synthetic data for various other domains. This research focuses solely on generating data for three basic emotions: positive, negative, and neutral. Future research could explore generating data for a broader range of nuanced emotions. Additionally, a novel, smaller T5 architecture could be developed using NAS to enhance the speed of synthetic data generation, as the current model requires approximately 5 hours and 21 minutes to generate 75,000 texts.

Author Contributions: Conceptualization: P. Sundarreson; Methodology: P. Sundarreson; Software: P. Sundarreson; Validation: P. Sundarreson; Formal analysis: P. Sundarreson.; Investigation: P. Sundarreson; Resources: P. Sundarreson; Data curation: P. Sundarreson.; Writing—original draft preparation: P. Sundarreson; Writing—review and editing: P. Sundarreson; Visualization: P. Sundarreson; Supervision: S. Kumarapathirage.; Project administration: P. Sundarreson.

Funding: This research received no external funding.

Data Availability Statement: The Hugging Face datasets (Yelp, SST-2, Bitcoin, Amazon, Finance News, IMDb, and Twitter) were utilized for this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] J. Luo, M. Bouazizi, and T. Ohtsuki, "Data Augmentation for Sentiment Analysis Using Sentence Compression-Based SeqGAN With Data Screening," *IEEE Access*, vol. 9, pp. 99922–99931, 2021, doi: 10.1109/ACCESS.2021.3094023.

[2] H. Q. Abonizio, E. C. Paraiso, and S. Barbon, "Toward Text Data Augmentation for Sentiment Analysis," *IEEE Trans. Artif. Intell.*, vol. 3, no. 5, pp. 657–668, Oct. 2022, doi: 10.1109/TAI.2021.3114390.

[3] S. A. Assefa, D. Dervovic, M. Mahfouz, R. E. Tillman, P. Reddy, and M. Veloso, "Generating synthetic data in finance," in *Proceedings of the First ACM International Conference on AI in Finance*, Oct. 2020, pp. 1–8. doi: 10.1145/3383455.3422554.

[4] M. Endres, A. Mannarapotta Venugopal, and T. S. Tran, "Synthetic Data Generation: A Comparative Study," in *International Database Engineered Applications Symposium*, Aug. 2022, pp. 94–102. doi: 10.1145/3548785.3548793.

[5] A. S. Imran, R. Yang, Z. Kastrati, S. M. Daudpota, and S. Shaikh, "The impact of synthetic text generation for sentiment analysis using GAN based models," *Egypt. Informatics J.*, vol. 23, no. 3, pp. 547–557, Sep. 2022, doi: 10.1016/j.eij.2022.05.006.

[6] P. Eigenschink, T. Reutterer, S. Vamosi, R. Vamosi, C. Sun, and K. Kalcher, "Deep Generative Models for Synthetic Data: A Survey," *IEEE Access*, vol. 11, pp. 47304–47320, 2023, doi: 10.1109/ACCESS.2023.3275134.

[7] G. Li, H. Wang, Y. Ding, K. Zhou, and X. Yan, "Data augmentation for aspect-based sentiment analysis," *Int. J. Mach. Learn. Cybern.*, vol. 14, no. 1, pp. 125–133, Jan. 2023, doi: 10.1007/s13042-022-01535-5.

[8] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, "Generation and evaluation of synthetic patient data," *BMC Med. Res. Methodol.*, vol. 20, no. 1, p. 108, Dec. 2020, doi: 10.1186/s12874-020-00977-1.

[9] C. F. Moreno-Garcia, C. Jayne, and E. Elyan, "Class-Decomposition and Augmentation for Imbalanced Data Sentiment Analysis," in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2021, pp. 1–7. doi: 10.1109/IJCNN52387.2021.9533603.

[10] J. Cui, Z. Wang, S.-B. Ho, and E. Cambria, "Survey on sentiment analysis: evolution of research methods and topics," *Artif. Intell. Rev.*, vol. 56, no. 8, pp. 8469–8510, Aug. 2023, doi: 10.1007/s10462-022-10386-z.

[11] A. Jadon and S. Kumar, "Leveraging Generative AI Models for Synthetic Data Generation in Healthcare: Balancing Research and Privacy," in *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, Jul. 2023, pp. 1–4. doi: 10.1109/SmartNets58706.2023.10215825.

[12] S. Akkaradamrongrat, P. Kachamas, and S. Sinthupinyo, "Text Generation for Imbalanced Text Classification," in *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Jul. 2019, pp. 181–186. doi: 10.1109/JCSSE.2019.8864181.

[13] J. Guan, R. Li, S. Yu, and X. Zhang, "A Method for Generating Synthetic Electronic Medical Record Text," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, pp. 1–1, 2019, doi: 10.1109/TCBB.2019.2948985.
[14] R. Gupta, "Data Augmentation for Low Resource Sentiment Analysis Using Generative Adversarial Networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 7380–7384. doi: 10.1109/ICASSP.2019.8682544.
[15] S. D. Sosun *et al.*, "Deep sentiment analysis with data augmentation in distance education during the pandemic," in *2022 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Sep. 2022, pp. 1–5. doi: 10.1109/ASYU56188.2022.9925379.
[16] C. Charitou, S. Dragicevic, and A. d'Avila Garcez, "Synthetic Data Generation for Fraud Detection using GANs." Sep. 26, 2021. [Online]. Available: http://arxiv.org/abs/2109.12546
[17] A. Kothare, S. Chaube, Y. Moharir, G. Bajodia, and S. Dongre, "SynGen: Synthetic Data Generation," in *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA)*, Nov. 2021, pp. 1–4. doi: 10.1109/ICCICA52458.2021.9697232.
[18] N. C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and L. Sweeney, "Privacy Preserving Synthetic Data Release Using Deep Learning," 2019, pp. 510–526. doi: 10.1007/978-3-030-10925-7_31.
[19] K. Fang, V. Mugunthan, V. Ramkumar, and L. Kagal, "Overcoming Challenges of Synthetic Data Generation," in *2022 IEEE International Conference on Big Data (Big Data)*, Dec. 2022, pp. 262–270. doi: 10.1109/BigData55660.2022.10020479.
[20] Y. Tao, R. McKenna, M. Hay, A. Machanavajjhala, and G. Miklau, "Benchmarking Differentially Private Synthetic Data Generation Algorithms." Dec. 16, 2021. [Online]. Available: http://arxiv.org/abs/2112.09238
[21] K. Pipalia, R. Bhadja, and M. Shukla, "Comparative Analysis of Different Transformer Based Architectures Used in Sentiment Analysis," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, Dec. 2020, pp. 411–415. doi: 10.1109/SMART50582.2020.9337081.
[22] P. Isaranontakul and W. Kreesuradej, "A Study of Using GPT-3 to Generate a Thai Sentiment Analysis of COVID-19 Tweets Dataset," in *2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Jun. 2023, pp. 106–111. doi: 10.1109/JCSSE58229.2023.10201994.
[23] A. Figueira and B. Vaz, "Survey on Synthetic Data Generation, Evaluation Methods and GANs," *Mathematics*, vol. 10, no. 15, p. 2733, Aug. 2022, doi: 10.3390/math10152733.
[24] Y. Shang, X. Su, Z. Xiao, and Z. Chen, "Campus Sentiment Analysis with GAN-based Data Augmentation," in *2021 13th International Conference on Advanced Infocomm Technology (ICAIT)*, Oct. 2021, pp. 209–214. doi: 10.1109/ICAIT52638.2021.9702068.
[25] T. Liesting, F. Frasincar, and M. M. Truşcă, "Data augmentation in a hybrid approach for aspect-based sentiment analysis," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, Mar. 2021, pp. 828–835. doi: 10.1145/3412841.3441958.
[26] J. Lee and J. Kim, "Improving Generation of Sentiment Commonsense by Bias Mitigation," in *2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Feb. 2023, pp. 308–311. doi: 10.1109/BigComp57234.2023.00061.
[27] X. Wang, S. Xue, J. Liu, J. Zhang, J. Wang, and J. Zhou, "Sentiment Classification Based on RoBERTa and Data Augmentation," in *2023 IEEE 9th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, Aug. 2023, pp. 260–264. doi: 10.1109/CCIS59572.2023.10263002.
[28] R. Xiang, E. Chersoni, Q. Lu, C. Huang, W. Li, and Y. Long, "Lexical data augmentation for sentiment analysis," *J. Assoc. Inf. Sci. Technol.*, vol. 72, no. 11, pp. 1432–1447, Nov. 2021, doi: 10.1002/asi.24493.
[29] A. Nazarizadeh, T. Banirostam, and M. Sayyadpour, "Using Group Deep Learning and Data Augmentation in Persian Sentiment Analysis," in *2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, Dec. 2022, pp. 1–5. doi: 10.1109/ICSPIS56952.2022.10044052.
[30] K.-H. Le Minh and K.-H. Le, "AirGen: GAN-based synthetic data generator for air monitoring in Smart City," in *2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI)*, Sep. 2021, pp. 317–322. doi: 10.1109/RTSI50628.2021.9597364.
[31] A. Ali and A. Said, "Generative Adversarial Networks (GANs): Models that can generate realistic synthetic data by training two competing neural networks." 2023. [Online]. Available: https://www.researchgate.net/publication/372649363_Generative_Adversarial_Networks_GANs_Models_that_can_generate_realistic_synthetic_data_by_training_two_competing_neural_networks
[32] A. Kiran and S. S. Kumar, "A Comparative Analysis of GAN and VAE based Synthetic Data Generators for High Dimensional, Imbalanced Tabular data," in *2023 2nd International Conference for Innovation in Technology (INOCON)*, Mar. 2023, pp. 1–6. doi: 10.1109/INOCON57975.2023.10101315.
[33] Z. Liu, J. Wang, and Z. Liang, "CatGAN: Category-Aware Generative Adversarial Networks with Hierarchical Evolutionary Learning for Category Text Generation," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 05, pp. 8425–8432, Apr. 2020, doi: 10.1609/aaai.v34i05.6361.
[34] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular data using Conditional GAN," Jun. 2019, [Online]. Available: http://arxiv.org/abs/1907.00503
[35] L. Bencke and V. P. Moreira, "Data augmentation strategies to improve text classification: a use case in smart cities," *Lang. Resour. Eval.*, Aug. 2023, doi: 10.1007/s10579-023-09685-w.
[36] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text Data Augmentation for Deep Learning," *J. Big Data*, vol. 8, no. 1, p. 101, Dec. 2021, doi: 10.1186/s40537-021-00492-0.
[37] J. Li, T. Tang, W. X. Zhao, and J.-R. Wen, "Pretrained Language Model for Text Generation: A Survey," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Aug. 2021, pp. 4492–4499. doi: 10.24963/ijcai.2021/612.
[38] A. Venkataramana, K. Srividya, and R. Cristin, "Abstractive Text Summarization Using BART," in *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, Oct. 2022, pp. 1–6. doi: 10.1109/MysuruCon55714.2022.9972639.
[39] H. Queiroz Abonizio and S. Barbon Junior, "Pre-trained Data Augmentation for Text Classification," 2020, pp. 551–565. doi: 10.1007/978-3-030-61377-8_38.

[40] A. Shuklin, D. Parygin, A. Gurtyakov, O. Savina, and N. Rashevskiy, "Synthetic News as a Tool for Evaluating Urban Area Development Policies," in *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, Oct. 2022, pp. 1–6. doi: 10.1109/ICEET56468.2022.10007405.

[41] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners." [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[42] X. Zheng, C. Zhang, and P. C. Woodland, "Adapting GPT, GPT-2 and BERT Language Models for Speech Recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec. 2021, pp. 162–168. doi: 10.1109/ASRU51503.2021.9688232.

[43] M. Binz and E. Schulz, "Using cognitive psychology to understand GPT-3," *Proc. Natl. Acad. Sci.*, vol. 120, no. 6, Feb. 2023, doi: 10.1073/pnas.2218523120.

[44] A. Mosallanezhad, K. Shu, and H. Liu, "Generating Topic-Preserving Synthetic News," in *2021 IEEE International Conference on Big Data (Big Data)*, Dec. 2021, pp. 490–499. doi: 10.1109/BigData52589.2021.9671623.

[45] N. El Houda Ouamane and H. Belhadef, "Deep Reinforcement Learning Applied to NLP: A Brief Survey," in *2022 2nd International Conference on New Technologies of Information and Communication (NTIC)*, Dec. 2022, pp. 1–5. doi: 10.1109/NTIC55069.2022.10100477.

[46] R. Behjati, E. Arisholm, M. Bedregal, and C. Tan, "Synthetic Test Data Generation Using Recurrent Neural Networks: A Position Paper," in *2019 IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)*, May 2019, pp. 22–27. doi: 10.1109/RAISE.2019.00012.

[47] R. Dos Santos, J. Aguilar, and M. D. R-Moreno, "A synthetic Data Generator for Smart Grids based on the Variational-Autoencoder Technique and Linked Data Paradigm," in *2022 XVLIII Latin American Computer Conference (CLEI)*, Oct. 2022, pp. 1–7. doi: 10.1109/CLEI56649.2022.9959918.

[48] S. Kamthe, S. Assefa, and M. Deisenroth, "Copula Flows for Synthetic Data Generation," Jan. 2021, [Online]. Available: http://arxiv.org/abs/2101.00598

[49] Y. Sei, J. A. Onesimu, and A. Ohsuga, "Machine Learning Model Generation With Copula-Based Synthetic Dataset for Local Differentially Private Numerical Data," *IEEE Access*, vol. 10, pp. 101656–101671, 2022, doi: 10.1109/ACCESS.2022.3208715.

[50] H. Liu *et al.*, "Using t-distributed Stochastic Neighbor Embedding (t-SNE) for cluster analysis and spatial zone delineation of groundwater geochemistry data," *J. Hydrol.*, vol. 597, p. 126146, Jun. 2021, doi: 10.1016/j.jhydrol.2021.126146.

[51] V. S. Kodiyala and R. E. Mercer, "Emotion Recognition and Sentiment Classification using BERT with Data Augmentation and Emotion Lexicon Enrichment," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2021, pp. 191–198. doi: 10.1109/ICMLA52953.2021.00037.

[52] R. Goyal, P. Kumar, and V. P. Singh, "A Systematic survey on automated text generation tools and techniques: application, evaluation, and challenges," *Multimed. Tools Appl.*, vol. 82, no. 28, pp. 43089–43144, Nov. 2023, doi: 10.1007/s11042-023-15224-0.

[53] N. Fatima, A. S. Imran, Z. Kastrati, S. M. Daudpota, and A. Soomro, "A Systematic Literature Review on Text Generation Using Deep Neural Network Models," *IEEE Access*, vol. 10, pp. 53490–53503, 2022, doi: 10.1109/ACCESS.2022.3174108.

[54] T. Iqbal and S. Qureshi, "The survey: Text generation models in deep learning," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2515–2528, Jun. 2022, doi: 10.1016/j.jksuci.2020.04.001.

[55] A. K. Pandey and S. S. Roy, "Natural Language Generation Using Sequential Models: A Survey," *Neural Process. Lett.*, vol. 55, no. 6, pp. 7709–7742, Dec. 2023, doi: 10.1007/s11063-023-11281-6.

[56] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagne, "DEAP: evolutionary algorithms made easy," *J. Mach. Learn. Res.*, vol. 13, pp. 2171–2175, 2012, [Online]. Available: https://www.jmlr.org/papers/volume13/fortin12a/fortin12a.pdf

[57] L. S. Hadla, T. M. Hailat, and M. N. Al-Kabi, "Comparative Study Between METEOR and BLEU Methods of MT: Arabic into English Translation as a Case Study," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 11, pp. 215–223, 2015, doi: 10.14569/IJACSA.2015.061128.