Cyberbully Detection Using BERT with Augmented Texts

Xiaoyu Guo
University of Arkansas
Fayetteville, Arkansas, USA
xsguo@uark.edu

Usman Anjum
University of Arkansas
Fayetteville, Arkansas, USA
uanjum@uark.edu

Jusin Zhan
University of Cincinnati
Cincinnati, Ohio, USA
zhanjt@ucmail.uc.edu

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020581

***Abstract*—Detecting cyberbullying in texts is an essential task as it curtails and identifies s ocial p roblems. I n t his p aper, we propose an architecture called Augmented BERT which combines both data augmentation techniques and BERT for detecting cyberbullying content in texts. Many techniques have been used in prior works to augment existing data for classification tasks and BERT had been applied in many text classification problems. However, there is a lack of annotated cyberbullying texts and obtaining annotated texts is hard and expensive. We propose to use various GAN-based and autoencoder-based data augmentation techniques to generate annotated data. The augmented texts can be used to fine-tune BERT. We choose to use HateBERT which is already pre-trained on abusive language to detect cyberbullying texts. Experimental results show an increased improvement over other cyberbullying detection models.**

***Index Terms*—data augmentation, cyberbully detection, natural language processing, sentence classification**

I. Introduction

With the rise of online forums platform like Twitter and Reddit, more and more people are openly expressing their opinions via the online forums. However, discussions may lead to disagreements and heated arguments, which can quickly turn into cyberbullying, which is defined a s a n aggressive, intentional act carried out by a group or individual, using electronic forms of contact, repeatedly and over time against victims who cannot easily defend themselves. [1].

Cyberbullying can take different forms, including flaming, harassment, denigration, impersonation, outing, boycotting, and cyberstalking [2], which can happen in different platforms like text messages, social media, or online games. There is a discrepancy between abusive language and hate speech. Some dataset treat both of them as cyberbully, while other dataset distinguish between them. [3] We used dataset with both definitions, b ecause w e w ant t o e xamine h ow robust our architecture will be when tested across different domains. There is also different techniques used for different language, and there is a challenge in uncommon language adapting the common language tools [4]. Text augmentation techniques are useful, because acquiring a sufficiently l arge d ataset f or every category on every platform with every language is impractical, but many developed models rely on a large number of human annotated training data, which is a slow and time-consuming process. In this paper, we present an architecture that uses text augmentation methods that generates similar texts as the training data. Then, we use both the generated text and the training data to fine-tune a pre-trained Bidirectional Encoder Representations from Transformers (BERT)-based classifier.

There are many techniques proposed for cyberbullying detection, but there is an inconsistency in the definition of cyberbullying text among different papers. Some paper claims that a sentence can be offensive, but not necessarily cyberbullying, while other paper claims that offensive language is a form of cyberbully. We found dataset with various definitions for model testing, because we want our model to be more robust and not limited to one specific definition. Some of the more popular techniques for cyberbully detection include deep neural network-based models like Convolutional Neural Network (CNN) or Long-Short Term Memory (LSTM), Naive Bayes Classifier, decision tree, SVM [5], and feature extractions [6]. The challenges for these models are slow training process, absence of multi-lingual classification, skewed dataset where very few data are marked as bullying, many offensive slang being mislabeled as hate speech or cyberbully, but the user may use those slang in daily communication, and misclassifying sentences as non-cyberbully when it does not have any offensive language.

For our proposed method, we assume that the training dataset is very small and obtaining a larger dataset is too costly. We want to firstly increase our dataset size by using various data augmentation techniques. We chose various GAN-based and autoencoder-based text augmentation techniques, because both are deep generative models and share some similarities between them, which we explain in section III-D. Then we use both the original testing data and the generated data to fine-tune HateBERT [7], a BERT-based classifier pre-trained with a large amount of English cyberbullying data. We want to explore how augmented data from different GANs and autoencoders can improve the classification performance.

The contributions of our work are as follows:

Formulation & Algorithm: We propose the *GAN Augmented BERT (`GAB`)* [1] model that utilizes different GAN text augmentation techinques and HateBERT to detect cyberbullying. To the best of our knowledge, no other works have focused

[1]https://github.com/gab624/GAB

on using augmented data from GAN to pre-train BERT and detect cyberbullying.

Generality: Our model is highly flexible and able to handle texts across different domains, including definition of cyberbullying and accurately detect cyberbullying text even with a small size of training data.

For the remaining of this paper, Section II discusses related works. Section III we introduce our methodology. In Section IV we present our analysis, evaluation and the results. We conclude in Section V.

II. Related Works

Augmented data has been used in prior literature for *data augmentation*. Data augmentation has been used in previous literature for image (e.g., face data augmentation in [8]), speech and natural language processing (NLP) [9]–[11] speech and time-series data to reduce overfitting [12], [13]. Data augmentation has also been implemented using agent-based models [14]. For generating texts, the most popular augmentation techniques use generative adversarial networks (GAN) [15]–[19] and autoencoders [20], [21]. These methods have modified GAN to deal with the discrete nature of text data. For example, SeqGAN [16] combines reinforcement learning and policy gradients to generate texts. In [22] and [19], a Gumbel-softmax distribution is implemented with GAN to generate texts.

The research on cyberbullying detection is fairly recent. Consequently, the methods implemented only focused on a limited definition of abuse. In fact, the definition of cyberbullying is not easy to define and there are multiple definitions that can be applied. Most of the literature has focused on using machine learning tools for cyberbullying detection. In [6] different machine learning classifiers were compared on cyberbullying detection using the Twitter datasets. Their results concluded that logistic regression was the best method. Other methods that have used machine learning for cyberbullying include [23]–[27]. A popular machine learning tool for cyberbullying detection was gated recurrent unit (GRU) which was used by [27] and [23]. Special filters were also designed to improve cyberbullying detection, e.g. [25] to reduce gender bias and [28] to improve detection of denigrating speech in cyberbullying.

In addition to machine learning models, transformation models like BERT [29] have also been used for cyberbullying classification. Pre-trained language models have been shown to have superior results. But these pre-trained models are designed for domain-specific applications. For our purpose we use HateBERT [7] a pre-trained BERT language model designed for the abusive language domain. Methods that have combined BERT and GAN for cyberbullying text classification was proposed by [30]. They used BERT to encode texts and implemented GAN as a classifier to identify type of abusive language.

There are models which incorporate data augmentation for skewed datasets. Methods like random mask and synonyms replacement had been use to solve the imbalance problem [31]. More complex methods like Markov Chains and LSTM had also been proposed [32]. There also exists a large dataset generated with GPT-3 [33] called TOXIGEN [34], which takes minority group into account.

III. Methodology

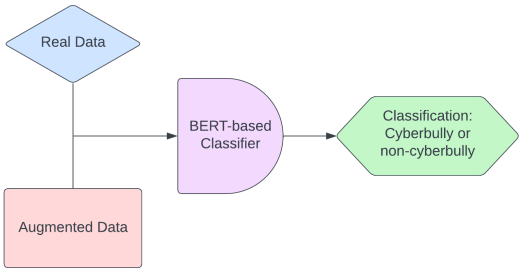

Fig. 1. GAB architecture

In this section, we describe this architecture in figure 1 that form the core of our model. There are two main components in our architecture. One is a text generating block, which includes SeqGAN [16], MaskGAN [15], RelGAN [19], Topic-Guided Variational Autoencoder [21], and Hybrid Concolutional VAE [35]. We use HateBERT as our classifier for measuring how well these augmentation techniques detect cyberbullying.

A. HateBERT

Bidirectional Encoder Representations from Transformers, or BERT, was proposed by Devlin et. al. [29]. Previously, language processing models read input text sequentially, either left-to-right or right-to-left. However, BERT reads the entire input text sequence all at once. This is possible because of the transformer that BERT is based on. Transformer is a deep learning model where every token from the output is connected to every token from the input with dynamically calculated weights.

BERT is trained with two NLP tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). MLM hides some words in a sentence, and BERT has to predict the masked words based on the context. NSP gives BERT two different sentences, and BERT needs to decide if the two sentences have a logical relationship, or if the two sentences are sequentially connected.

In our model, we chose HateBERT [7] as our main classifier. HateBERT was trained with over 1.49 million messages from Reddit with Masked Language Model (MLM) objective. MLM is also used in the fine-tuning process. With MLM, we are inputting a sequence with masked tokens to transformer-based BERT, and expecting BERT to complete the sequence, and the completed sequence needs to be as close to the original sequence as possible. After BERT's attempt of completing the sentence, we will calculate the loss function and the required gradient changes, which BERT will utilize to optimize the weights of the model. HateBERT is more robust than many other text classification techniques [7]. Since HateBERT is

already pre-trained on the abusive language phenomenon, and we can further fine-tune HateBERT with our testing dataset via MLM, HateBERT is a more suitable option for detection of cyberbullying text.

B. Generative Adversarial Networks

Generative adversarial networks, or GAN, was proposed in 2014 by Goodfellow et al [36]. The framework of GAN includes two models: a generator G and a discriminator D. The generator is trained by learning the statistical distribution of the training data, and generates similar data to the training data. Then the original training data and the generated data is fed into the discriminator randomly, in which the discriminator needs to classify if the data is from the training set or generated by G. Essentially, D and G plays a two-player minimax game, with the value function

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

$D(x)$ is the probability determined by the discriminator that x is from the real dataset. Given a latent vector z from some representation space, $G(z)$ is the data generated by the discriminator. From the value function, the discriminator wants to maximize the accuracy of its classification of the input data as "real" or "generated" data, while the generator wants to minimize the possibility that the discriminator identifies the generated data accurately, instead wanting to fool the discriminator into believing that the generated data is from the real dataset.

GAN is extremely useful in image generation, because the training of the generator relies on stochastic gradient. The discriminator and the generator are trained by descending their respective stochastic gradient. This implies that all layers of the GAN must be differentiable, which is feasible when working with continuous data, such as images.

C. Text Augmentation GANs

For text generation, because the data is discrete, some layers may not be differentiable, and the gradient updates become a problem. There are many different text-generating GAN models that have attempted to overcome this problem [15]–[19]. We chose SeqGAN, MaskGAN, and RelGAN, because the these GANs are the most widely used text generating GANs, and also have strong performances [37]–[39].

MaskGAN [15] utilizes actor-critic method. It treats text generation as a fill-in-the-blank task. The generator has an encoding module, which reads in a masked sequence; and a decoding module, which computes the missing tokens from the masked sequence via the encoder hidden state. Adversarially, the Long-Short Term Memory, or LSTM-based discriminator is given the filled-in sequence from the decoder of the generator and the context sequence, which determines which tokens in the original sequence will remain unmasked. The discriminator computes a scalar, which is the probability of each token being the real token, given the true context of the masked sequence. The reward function is the logarithm of the discriminator estimated scalar. There is an additional critic network that estimates the value function that returns the discounted total return value of the filled-in sequence, which is used to train the discriminator. The discount factor used in the critic network is determined at each position in the sequence.

SeqGAN [16] is represented as a reinforcement learning problem, where the state is the set of text generated so far, and an action that is the next work to choose in a sentence. The generator in SeqGAN chooses a word at each time step. When the end of the sentence action is reached, a reward is provided by the discriminator network. SeqGAN is trained using policy gradient. The policy function defines a probability distribution over all actions given the current state. The text is generated sampled from a set of actions according to the distribution returned by the policy function. The generator is implemented using LSTM with the objective of generating a sequence from a start state such that the expected end reward is maximized.

Relational GAN [19], or RelGAN, includes a generator based on relational memory, a discriminator with multiple embedded representations, and is trained with Grumbel-Softmax relaxation. The generator considers a fixed set of memory slots, then uses self-attention mechanism to prompt interactions between the memory slots. The generator is trained with the standard MLE before the adversarial training. The discriminator for RelGAN has multiplied embedded representations for each input sentence, meaning that each representation will pass through a CNN-based classifier independently to get an individualized score, and the average of all scores is used to update the generator. The loss function used by RelGAN estimates the average probability that the embedded representations of the real sentences are more realistic than the representations of generated sentences. The author claims that RelGAN produces more realistic and diverse text than other text generating GANs.

Text-generating GANs face a challenge called mode collapse, when the generator only outputs a single type of output, which becomes repetitive and uninformative. Fill-in-the-blank method can help with mode collapse to a degree, which makes MaskGAN more stable.

From the randomly selected examples in Table I generated based on the same set of training data, we can see that RelGAN does generate more realistic text than the other two, and RelGAN does have the highest BLEU score among all the methods we tested.

D. Text Augmentation Autoencoders

To investigate further into text generating models, we also took consideration of variational autoencoder [VAE], because VAE is another deep generative model with two components. Both GAN and VAE also utilizes unsupervised learning. Instead of a generator and a discriminator, a VAE has an encoder, which compresses the input data into a latent space, and a decoder, which tries to decode the input from the latent space to the original input data.

	generated text
MaskGAN	a plane is waiting for his fat toilet to get on board his angry bird skill has taken over the world you can be the president with that overwhelming ego
SeqGAN	a cup of your dad is ugly stupid kids hanging from the tree decorated with christmas lights his tall pathetic virgin across the street
RELGAN	he said you're a bad person you lost your job, which is why you're homeless you're so stinky i want to kill myself

TABLE I
GENERATED TEXT EXAMPLES

We chose Topic Guided VAE [21] and Hybrid Convolutional VAE [35] for comparison. Different from GAN, which generates text adversarially between the generator and the discriminator, autoencoder utilizes an encoder and a decoder. The encoder will take in an input and compress to a much smaller dimension, and the decoder will try to reconstruct the input based on the compressed information. Because the encoder encodes the input into a smaller dimension, autoencoders are useful when learning features among the input data.

E. Comparison of Different Text Augmentation Techniques

One of the most popular metrics to evaluate text augmentation techniques is the BLEU score [40]. BLEU score calculates how close a generated sentence is to a reference sentence. In other words, it examines how many words appeared in both the generated sentence and the reference sentence. We use BLEU-2, which is the average of BLEU score evaluated with 1-gram and 2-gram. 1-gram means we only compare one word at a time, and 2-gram means we evaluate two consecutive words at a time. The reference sentences are randomly selected from the training set. For individual BLEU scores, we compare each generated sentence with all the reference sentences. To get the cumulative BLEU score, we simply take the average of all the individual BLEU scores. However, BLEU score has its own limitations, which we discuss in the later section.

As presented in Table II, we calculated the BLEU score from each text augmentation techniques. RelGAN produced the best BLEU score, which means it is the most realistic and human-like text generating technique among the five methods. However, we will see later that realistic generated text does not necessarily improve cyberbullying detection the most.

Methods	BLEU
SeqGAN	0.503
MaskGAN	0.494
RelGAN	**0.531**
TGVAE	0.479
Hybrid CVAE	0.519

TABLE II
TEXT GENERATING METHODS COMPARISON

IV. ANALYSIS AND EVALUATION

A. Datasets

For the experiments, we picked five different datasets:

1) **HaSpeeDe:** Italian hate speech dataset HaSpeeDe from the 2018 EVALITA Competition, and the original data is included in [30]. The classification is "hate speech" or "not hate speech". This dataset has 3,000 entries.
2) **Twitter-1:** English Twitter dataset from [41], where the data is classified as "hate speech", "offensive but not hate speech", or "neither". This dataset has 24,783 entries.
3) **YouTube:** English YouTube comments dataset from [42]. The data is classified as either "bullying" or "non-bullying". This dataset has 3,464 entries.
4) **Twitter-2:** English Twitter dataset from [43]. The data is classified as either "cyberbullying" or "non-cyberbullying". This dataset has 46,017 entries. (https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification)
5) **Kaggle:** English Kaggle parsed dataset from https://data.mendeley.com/datasets/jf4pzyvnpj/1. The text is classified as "cyberbullying" or "non-cyberbullying". This dataset has 8,799 entries.

Dataset	Classes	# per class
HaSpeeDe	hate, not hate	972, 2028
Twitter-1	hate, offensive but not hate, neither	1430, 19190, 4163
YouTube	cyberbully, non-cyberbully	417, 3047
Twitter-2	cyberbully, non-cyberbully	7945, 38072
Kaggle	cyberbully, non-cyberbully	2806, 5993

TABLE III
DATASET USED IN EXPERIMENTS

B. Evaluation Metrics

For measuring our model's performance, we calculate both the accuracy and the F1-score. Accuracy measures how many times a model predicted correctly.

$$\text{Accuracy} = \frac{\text{correct predictions}}{\text{all predictions}}$$

However, accuracy has its limitations. If our dataset is imbalanced, say we have 90% cyberbullying text and 10% non-cyberbullying text. Our model may have a 90% accuracy, but is a very bad model because it predicts everything as cyberbullying. To better evaluate our model, we also use the F1-score, which examines the types of errors a model made.

The F1-score ($F1$) is defined as:

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{1}$$

where TP are the true positives, FP are the false positives and FN are the false negative.

C. Pre-Processing

Before we can train using GAN augmented texts, we need to pre-process the datasets. We split all of our datasets by a 70/30 ratio, where 70% of each dataset is used as the training set, and the remaining 30% is the testing set.

For every sentence in the training set, we firstly remove any leading or trailing spaces. We then split each sentence into a list of words, and turn each word into its integer representation. Then the list of integer representations will be appended to the comprehensive list of all the sentences in the dataset. The comprehensive list will be shuffled and split into different batches to train the generator and the discriminator alternately.

D. Comparison of Different Augmentation Techniques

To test how each text augmentation methods improve cyberbullying detection, we use the generated text from each method to further fine-tune the HateBERT, which was already fine-tuned with the testing data from Twitter-2. From Table V, we can see that even though RelGAN has the highest BLEU score, SeqGAN actually achieves the best accuracy and F-1 score. This infers that there is no correlation between good text augmentation techniques and improvement on cyuberbully detection. Because SeqGAN improves HateBERT the most, we decide to run SeqGAN on more datasets, and see if HateBERT fine-tuned with SeqGAN augmented text is better than other architectures.

E. SeqGAN for Cyberbullying Classification

In this section we develop the following algorithm for further investigae SeqGAN. We use SeqGAN to generate more data similar to our dataset for fine-tuning HateBERT. We will read each data from the pre-processed lists, and append the data to one of the two lists, one called "positive_examples" and the other "negative_examples". We then give each data in the lists a positive or negative label, and split the lists into different batches after shuffling the lists. Then we can use the different batches to train the generator and the discriminator alternately, until we reach a convergence.

We define a sequence as $Y_{1:T} = (y_1, ..., y_t, ..., y_T), y_t \in \mathcal{Y}$, where $\mathcal{Y}$ is the vocabulary of candidate words. Equation 2 is a Monte-Carlo (MC) method for eavaluting the action-value function $Q(s, a)$ of a sequence, where $MC^{G_\theta}(Y_{1:t;N}) = Y^1_{1:T}, ..., Y^N_{1:T}$ are the MC rollouts with policy G_θ and $Y^n_{1:t} = y_1, ..., y_t$. The D_ϕ is the discriminator which is used to implement the reward function and $D_\phi = Q(a = y_T, s = Y_{1:T-1})$. The discriminator can be dynamically updated to further improve the generator iteratively. It can be re-trained as more sequences are generated. The discriminator is implemented using convolutional neural network (CNN) and is retrained using Equation 4. The generator parameter is updated as Equation 3, where α is the learning rate.

Algorithm 1: GAB for Cyberbully Detection

Data: Input Sentences
Result: Outputs classes
Initialization: Pre-processing texts based on section 4.1
SeqGAN Initialization using training set:
repeat
  for *generator training* **do**
    generate a sequence T
    for *t in T* **do**
      compute Equation

$$Q^{G_\theta}_{D_\phi}(S = Y_{1:t-1,a=y_t}) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y^n_{1:T}) \in MC^{G_\theta}(Y_{1:t}; N) & t < T \\ D_\phi(Y_{1:t}) & t = T \end{cases} \quad (2)$$

    end
    update generator via Equation

$$\theta_{t+1} = \theta_t + \alpha \Delta_\theta J(\theta) \quad (3)$$

  end
  for *discriminator training* **do**
    train discriminator with Equation

$$min\Big[\mathbb{E}_Y[logD_\phi(Y)] - \mathbb{E}_{Y\sim G_\theta}[log(1 - D_\phi(Y))]\Big] \quad (4)$$

    for k epochs
  end
until *SeqGAN converges*;
SeqGAN generates sequences $\hat{Y}_{1:N}$
concatenate $\hat{Y}_{1:N}$ with the original dataset
fine-tune HateBERT
classify on the test set using HateBERT

F. Experimental Results

The hyperparameters for SeqGAN and HateBERT can be found in Appendix A. We chose these hyperparameters because they are recommended by the authors of these models. All experiments are ran on a Windows 10 Enterprise with 128 GB RAM, i9-9900K CPU at 3.6GHz and Nvidia Quadro RTX 4000 GPU.

For comparison, we ran experiments on all five datasets with BERT [29] only, HateBERT [7] only, the GANBERT model [44], and our proposed model, GAB. The results are listed in Table IV. We chose GANBERT, because it is another architecture that utilized both GAN and BERT. In GANBERT, BERT is used as a data processor, which processes the real data, and feed the processed data into the discriminator along with the data from the generator.

We see that GAB produces the best F1-scores in three datasets, and the best accuracy scores in two datasets. Both YouTube and Kaggle dataset has relatively fewer entries than the other datasets, which is where our model excels. Twitter-2

	HaSpeeDe		Twitter-1		YouTube		Twitter-2		Kaggle	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
BERT	0.453	0.417	0.501	0.598	0.410	0.488	0.705	0.692	0.496	0.500
HateBERT	0.316	0.379	0.682	**0.798**	0.468	0.503	**0.774**	**0.801**	0.503	0.540
GANBERT	**0.555**	**0.619**	0.548	0.602	0.427	0.504	0.630	0.713	0.414	0.492
GAB	0.519	0.583	**0.702**	0.780	**0.507**	**0.584**	0.739	0.791	**0.516**	**0.593**

TABLE IV
THE AVERAGE F1 AND ACCURACY SCORES FOR THE FIVE DATASETS LISTED ABOVE.

Methods	Accuracy	F1 score
SeqGAN	**0.739**	**0.791**
MaskGAN	0.702	0.694
RelGAN	0.671	0.738
TGVAE	0.719	0.705
Hybrid 0.519	0.69	0.683

TABLE V
TEXT GENERATING METHODS COMPARISON

has the most data entries and our model loses our advantages from data augmentation to HateBERT. HaSpeeDe is an Italian dataset, and the GANBERT model utilizes UmBERTo [45], [46], an Italian BERT-based transformer, which has more advantages processing HaSpeeDe than our English-pre-trained HateBERT. It is also worth noting that BERT performed better for HaSpeeDe than HateBERT. Our model outperforms HateBERT in F1 score for Twitter-1, but has lower accuracy. This tells us that HateBERT has more accurate predictions than GAB, but GAB performs better regarding due to low false positives and false negatives.

V. CONCLUSION

In this paper, we realized that acquiring a large amount of human-annotated training data for cyberbully detection is costly and time-consuming. We assume that our training set is limited, so we proposed an architecture where we first generate more data similar to the training data. Then we can use the generated data along with the original training data to further fine-tune HateBERT. This architecture performs better than simply using BERT or HateBERT. We also explored how different data augmentation techniques improve the overall text classification performance.

In the future, we want to examine more text augmentation techniques, including GTP-3, domain adaptation, and more GAN- and autoencoder-based techniques. We want to further explore how our proposed architecture perform across datasets with different definitions of cyberbullying. We would like to improve our architecture to have a modular architecture, where we can swap between different data augmentation techniques and classification techniques. We also plan to use GAN to generate feature-specific data, which will make the generated data more robust and diverse.

A. Limitations

When deciding which text generation technique we want to use in our architecture, we used the BLEU score metrics. However, in 2018, Post claimed in his paper that it is difficult to compare BLEU cores across papers [47]. One of the main problem is under-specifying parameters for BLEU calculation. In our method, we computed 1-gram and 2-gram BLEU score, but other papers may use only other methods to calculate BLEU score, and comparing BLEU scores across papers with different calculation methods are nonsensical. Another problem with BLEU score is that we are comparing how many words from the generated data also appeared in the original data, which is costly to compute, and may not produce an accurate score. A generated sentence can suit the requirements perfectly, but still receive a low BLEU score, because it uses some words that did not appear in the original dataset.

Evaluating BLEU scores also takes a longer time than the actual training of our model. This is because we compare each generated sentence to a large amount of original data entries. There exists a lack of uniform computational evaluation metrics for text generation, but there are other evaluation methods. One such method is negative log-likelihood(NLL) introduced in SeqGAN, which evaluates how good the data is fitted by the oracle language model. Further alterations of NLL are also options for future exploration. For example, one alteration NLL_{div} [48] evaluates the repeatability of the generated samples.

We also realize that HateBERT is English-trained, which results in our model performing poorly in datasets of other languages. This is a major challenge in the NLP field, because different languages have different characteristics and pre-training models can miss the unique characteristics of a foreign language. The HateBERT was pre-trained on an English dataset and it performs worse on the Italian dataset. HateBERT was also pre-trained specifically for hate speech detection, so it will perform way worse on other tasks, like feature extraction. Pre-training is very time-consuming and computationally costly, so it is infeasible to pre-train BERT for every distinct task and dataset. There are some publications on multilingual AMR-to-text generation [49], [50], where the sentences are generated from some Abstract Meaning Representation graphs. We can construct AMR graphs from existing English datasets and produce more sentences in other languages for training the language-specific classifiers. However, balancing the training cost and classification accuracy remains a dilemma in text classification.

VI. Acknowledgements

This work was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award P20GM139768, and the Arkansas Integrative Metabolic Research Center at the University of Arkansas. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

This work was also supported by the National Science Foundation under Award No. OIA-1946391. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Science Foundation.

References

[1] P. K. Smith, J. Mahdavi, M. Carvalho, S. Fisher, S. Russell, and N. Tippett, "Cyberbullying: Its nature and impact in secondary school pupils," *Journal of child psychology and psychiatry*, vol. 49, no. 4, pp. 376–385, 2008.

[2] D. Siegle, "Cyberbullying and sexting: Technology abuses of the 21st century," *Gifted child today*, vol. 33, no. 2, pp. 14–65, 2010.

[3] T. Davidson, D. Bhattacharya, and I. Weber, "Racial bias in hate speech and abusive language detection datasets," *arXiv preprint arXiv:1905.12516*, 2019.

[4] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in social networks: a survey on multilingual corpus," in *6th International Conference on Computer Science and Information Technology*, vol. 10, 2019, pp. 10–5121.

[5] M. Dadvar and K. Eckert, "Cyberbullying detection in social networks using deep learning based models; a reproducibility study," *arXiv preprint arXiv:1812.08046*, 2018.

[6] A. Muneer and S. M. Fati, "A comparative analysis of machine learning techniques for cyberbullying detection on twitter," *Future Internet*, vol. 12, no. 11, p. 187, 2020.

[7] T. Caselli, V. Basile, J. Mitrović, and M. Granitzer, "Hatebert: Retraining bert for abusive language detection in english," *arXiv preprint arXiv:2010.12472*, 2020.

[8] X. Wang, K. Wang, and S. Lian, "A survey on face data augmentation for the training of deep neural networks," *Neural Computing and Applications*, pp. 1–29, 2020.

[9] X. Dai and H. Adel, "An analysis of simple data augmentation for named entity recognition," *arXiv preprint arXiv:2010.11683*, 2020.

[10] U. Anjum, "Localization of events using underdeveloped microblogging data," Ph.D. dissertation, University of Pittsburgh, 2022.

[11] U. Anjum, V. Zadorozhny, and P. Krishnamurthy, "A deep learning framework for event detection in augmented twitter data," *Available at SSRN 4124071*.

[12] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[13] Q. Wen, L. Sun, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[14] U. Anjum, V. Zadorozhny, and P. Krishnamurthy, "Tbam: Towards an agent-based model to enrich twitter data," in *18th ISCRAM Conference Proceedings*. Virginia Tech., ISCRAM Blacksburg, VA (USA), 2021.

[15] W. Fedus, I. Goodfellow, and A. M. Dai, "Maskgan: better text generation via filling in the_," *arXiv preprint arXiv:1801.07736*, 2018.

[16] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[17] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun, "Adversarial ranking for language generation," *Advances in neural information processing systems*, vol. 30, 2017.

[18] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[19] W. Nie, N. Narodytska, and A. Patel, "Relgan: Relational generative adversarial networks for text generation," in *International conference on learning representations*, 2018.

[20] D. Donahue and A. Rumshisky, "Adversarial text generation without reinforcement learning," *arXiv preprint arXiv:1810.06640*, 2018.

[21] W. Wang, Z. Gan, H. Xu, R. Zhang, G. Wang, D. Shen, C. Chen, and L. Carin, "Topic-guided variational autoencoders for text generation," *arXiv preprint arXiv:1903.07137*, 2019.

[22] M. J. Kusner and J. M. Hernández-Lobato, "Gans for sequences of discrete elements with the gumbel-softmax distribution," *arXiv preprint arXiv:1611.04051*, 2016.

[23] A. Kumar and N. Sachdeva, "A bi-gru with attention and capsnet hybrid model for cyberbullying detection on social media," *World Wide Web*, pp. 1–14, 2021.

[24] M. Dadvar and K. Eckert, "Cyberbullying detection in social networks using deep learning based models," in *International conference on big data analytics and knowledge discovery*. Springer, 2020, pp. 245–255.

[25] J. H. Park, J. Shin, and P. Fung, "Reducing gender bias in abusive language detection," *arXiv preprint arXiv:1808.07231*, 2018.

[26] E. Raisi and B. Huang, "Cyberbullying detection with weakly supervised machine learning," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, 2017, pp. 409–416.

[27] M. Alotaibi, B. Alotaibi, and A. Razaque, "A multichannel deep learning framework for cyberbullying detection on social media," *Electronics*, vol. 10, no. 21, p. 2664, 2021.

[28] S. R. Sangwan and M. Bhatia, "D-bullyrumbler: a safety rumble strip to resolve online denigration bullying using a hybrid filter-wrapper approach," *Multimedia Systems*, pp. 1–17, 2020.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[30] C. Breazzano, D. Croce, and R. Basili, "Mt-gan-bert: Multi-task and generative adversarial learning for sustainable language processing," 2021.

[31] M. Ibrahim, M. Torki, and N. El-Makky, "Imbalanced toxic comments classification using data augmentation and deep learning," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2018, pp. 875–878.

[32] 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)*. IEEE, 2019, pp. 181–186.

[33] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[34] T. Hartvigsen, S. Gabriel, H. Palangi, M. Sap, D. Ray, and E. Kamar, "Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection," *arXiv preprint arXiv:2203.09509*, 2022.

[35] S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," *arXiv preprint arXiv:1702.02390*, 2017.

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[37] B. Li, Y. Hou, and W. Che, "Data augmentation approaches in natural language processing: A survey," *AI Open*, 2022.

[38] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of data augmentation approaches for nlp," *arXiv preprint arXiv:2105.03075*, 2021.

[39] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *Journal of big Data*, vol. 8, no. 1, pp. 1–34, 2021.

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

[41] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ser. ICWSM '17, 2017, pp. 512–515.

[42] M. Dadvar, D. Trieschnigg, and F. d. Jong, "Experts and machines against bullies: A hybrid approach to detect cyberbullies," in *Canadian conference on artificial intelligence*. Springer, 2014, pp. 275–281.

[43] J. Wang, K. Fu, and C.-T. Lu, "Sosnet: A graph convolutional network approach to fine-grained cyberbullying detection," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 1699–1708.
[44] D. Croce, G. Castellucci, and R. Basili, "Gan-bert: Generative adversarial learning for robust text classification with a bunch of labeled examples," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 2114–2119.
[45] B. Magnini, A. Cappelli, E. Pianta, M. Speranza, V. Bartalesi Lenzi, R. Sprugnoli, L. Romano, C. Girardi, and M. Negri, "Annotazione di contenuti concettuali in un corpus italiano: I - cab," in *Proc.of SILFI 2006*, 2006.
[46] B. Magnini, E. Pianta, C. Girardi, M. Negri, L. Romano, M. Speranza, V. B. Lenzi, and R. Sprugnoli, "I - cab: the italian content annotation bank." in *LREC*. Citeseer, 2006, pp. 963–968.
[47] M. Post, "A call for clarity in reporting bleu scores," *arXiv preprint arXiv:1804.08771*, 2018.
[48] Z. Liu, J. Wang, and Z. Liang, "Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8425–8432.
[49] A. Fan and C. Gardent, "Multilingual amr-to-text generation," *arXiv preprint arXiv:2011.05443*, 2020.
[50] L. F. Ribeiro, J. Pfeiffer, Y. Zhang, and I. Gurevych, "Smelting gold and silver for improved multilingual amr-to-text generation," *arXiv preprint arXiv:2109.03808*, 2021.

APPENDIX

Hyperparameters	**Values**
Learning rate	1e-5
Training Epoch	5
Adam epsilon	1e-8
Max sequence length	100
Batch size	32
Num. warmup steps	0

TABLE VI
HYPERPARAMETERS FOR FINE-TUNING HATEBERT

Hyperparameters	**Values**
generator training	30
discriminator training	1
training epochs [k]	30

TABLE VII
HYPERPARAMETERS FOR SEQGAN