HateGAN: Adversarial Generative-Based Data Augmentation for Hate Speech Detection

Rui Cao
Singapore Management University
ruicao.2020@phdcs.smu.edu.sg

Roy Ka-Wei Lee
Singapore University of Technology and Design
roy_lee@sutd.edu.sg

Abstract

Academia and industry have developed machine learning and natural language processing models to detect online hate speech automatically. However, most of these existing methods adopt a supervised approach that heavily depends on labeled datasets for training. This results in the methods' poor detection performance of the hate speech class as the training datasets are highly imbalanced. In this paper, we propose HateGAN, a deep generative reinforcement learning model, which addresses the challenge of imbalance class by augmenting the dataset with hateful tweets. We conduct extensive experiments to augment two commonly-used hate speech detection datasets with the HateGAN generated tweets. Our experiment results show that HateGAN improves the detection performance of the hate speech class regardless of the classifiers and datasets used in the detection task. Specifically, we observe an average 5% improvement for the hate class F1 scores across all state-of-the-art hate speech classifiers. We also conduct case studies to empirically examine the HateGAN generated hate speeches and show that the generated tweets are diverse, coherent, and relevant to hate speech detection.

1 Introduction

Motivation. The sharp increase in online hate speeches has raised concerns globally (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018). The spread of hate speech in social media has not only sowed discord among individuals or communities online but also resulted in violent hate crimes (Williams, 2019; Relia et al., 2019; Mathew et al., 2019). Therefore, it is a pressing issue to detect and curb hate speech in online social media. Major social media platforms such as Facebook and Twitter have made great efforts to combat the spread of hate speech in their platforms (Times, 2019; Bloomberg, 2019). Researchers have also proposed many traditional and deep learning hate speech classification methods to detect hate speeches in online social media automatically (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018). Most of these existing methods adopt a supervised approach that heavily depends on labeled datasets for training. This results in the methods' poor detection performance of the hate speech class as the training datasets are highly imbalanced (Waseem and Hovy, 2016; Davidson et al., 2017).

A potential solution to address the challenges of imbalance datasets is to perform data augmentation for the class with fewer training samples. For instance, perturbing replicas of data samples using noise injection or attribute modification techniques have been successful in other domains such as image and sound classification tasks (Shorten and Khoshgoftaar, 2019; Tran et al., 2017; Dong et al., 2016; Keren et al., 2016; Salamon and Bello, 2017). Nevertheless, these techniques are not transferable to text as they would break the text's syntax and alter the semantics of the original sentences. There are also very few works that have explored improving hate speech detection performance using data augmentation, and their results have shown limited improvement on automatic hate speech detection (Rizos et al., 2019).

Research Objectives. In this paper, we aim to fill the research gaps by proposing HateGAN, a novel controlled text generation method to generate diverse and relevant short hate speeches to augment existing social media hate speech datasets. At a high level, HateGAN adopts a reinforcement learning-based

Proceedings of the 28th International Conference on Computational Linguistics, pages 6327–6338
Barcelona, Spain (Online), December 8-13, 2020

generative adversarial network architecture. The generative adversarial network component ensures that the generated text stays relevant and syntactically close to the original dataset. Inspired by the SeqGAN model (Yu et al., 2017), the reinforcement learning-based component encourages the generator to generate text that is more hateful by introducing a reward policy gradient to guide its generation. Specifically, we include a pre-trained toxicity scorer that scores a given text on six dimensions of toxic content. The computed scores are used as rewards and feedback signals to guide the generation of hateful text.

Contributions.[1] Our main contributions in this work consist of the following.

- We propose a novel deep learning model called HateGAN, which adopts a reinforcement learning-based generative adversarial network architecture to generate hate speech for data augmentation.
- We conduct extensive experiments to augment two commonly-used hate speech detection datasets. Our experiment results show that HateGAN improves the detection performance of the hate speech class regardless of the classifiers and datasets used in the detection task. Specifically, we observe an average 5% improvement for the hate class F1 scores across all state-of-the-art hate speech classifiers.
- We conduct empirical analyses on the generated hate speech and show that the generated texts are diverse, coherent, and relevant to hate speech detection.

2 Related Work

Automatic detection of hate speech has received considerable attention from data mining, information retrieval, and natural language processing (NLP) research communities. Interest in this field has increased with the proliferation of social media and social platforms (Fortuna and Nunes, 2018; Schmidt and Wiegand, 2017). Traditional models (Chen et al., 2012; Waseem and Hovy, 2016; Waseem, 2016; Nobata et al., 2016; Chatzakou et al., 2017; Davidson et al., 2017) and deep learning models (Djuric et al., 2015; Mehdad and Tetreault, 2016; Gambäck and Sikdar, 2017; Badjatiya et al., 2017; Park and Fung, 2017; Gröndahl et al., 2018; Zhang et al., 2018; Arango et al., 2019; Founta et al., 2019) have been developed to detect hate speech in social media. Most of these existing methods adopt a supervised approach that heavily depends on labeled datasets for training, which is a challenge as existing hate speech datasets are highly imbalanced. For example, Waseem and Hovy (2016) collected a Twitter dataset and manually annotated the sexism and racism tweets. Davidson et al. (2017) performed similar annotation on a larger Twitter corpus, where the researchers differentiated *offensive* from *hate* tweets. However, both datasets are highly imbalanced; only about 30% of the tweets in (Waseem and Hovy, 2016) are labeled as sexism or racism, and less than 12% of the tweets in (Davidson et al., 2017) are labeled as hateful.

Data augmentation methods have been explored to address the imbalance datasets challenge in supervised classification tasks. For example, noise injection or attribute modification techniques were commonly applied to generate synthetic data for image and sound classification tasks (Shorten and Khoshgoftaar, 2019; Tran et al., 2017; Dong et al., 2016; Keren et al., 2016; Salamon and Bello, 2017). Nevertheless, it is challenging to apply such techniques on text due to the categorical nature of words and the sequential nature of text. In recent years, generative adversarial network models such as SeqGAN (Yu et al., 2017) and LeakGAN (Guo et al., 2018) have been proposed to generate text. However, these approaches suffer from high training variance and mode collapse. Controlled text generation techniques were also explored to perform data augmentation (Malandrakis et al., 2019). For instance, Variational Autoencoder (VAE) was applied in text generation (Bowman et al., 2016). The VAE models consist of an encoder that maps each sample to a latent representation and a decoder that generates samples from the latent space and a variational attribute (Kingma and Welling, 2013). The variational attribute is able to add diversity to the generated output sequences. The Conditional VAE (CVAE) (Kingma et al., 2014) was proposed to incorporate stochastic latent variables to improve the generation of diverse and relevant text. In this paper, we propose HateGAN, which a novel a reinforcement learning-based generative adversarial network to generate short hate speeches. Specifically, the policy gradient component

[1] Code:https://gitlab.com/bottle_shop/safe/hategan

in HateGAN guides the text generation process, which creates diverse and relevant short hate speeches. The generated hate speeches are subsequently used for data augmentation to improve automatic hate speech detection.

There are very few works that explored data augmentation in hate speech detection. In a recent work, Rizos et al. (2019) explored three kinds of data augmentation for hate speech detection. The researchers had explored (1) substituting the words in text, (2) swapping the word positions, and (3) the neural generation method using Recurrent Neural Network (RNN) (Sutskever et al., 2011). Each of these methods has its limitations. For instance, word substitution is challenging for hate speech generation as certain words may inherently contain hateful or offensive semantics. Furthermore, new lexicons may be created in the fast-evolving social media, adding challenges to find reasonable semantically similar words for substitution. It is also challenging to swap words position while maintaining the coherence of the sentences. Lastly, the neural generation method used in the study was generic and rudimentary. In this paper, we also adopt a neural generation approach to augment data for hate speech detection. However, unlike the neural generation method in (Rizos et al., 2019), our proposed HateGAN model is specifically designed to generate high quality and diverse short hate speeches.

3 Adversarial Generative-Based Model

Figure 1 illustrates the overall architecture of our proposal HateGAN model. It follows a similar process of reinforcement learning-based sequence generation proposed in (Yu et al., 2017). The discriminator is trained to guide the generator to synthesize tweets that are indistinguishable from the real tweets. The "realisticness" of the synthesized tweets is measured and output as *realistic scores*. As our goal is to improve the performance of hate speech detection by data augmentation, we aim to generate more hateful tweets. Therefore, we pre-trained a toxicity scorer that quantifies the hatefulness of the synthesized tweets as *hate scores*. The *realistic scores* and *hate scores* are subsequently used as rewards to guide and update the parameters in the generator for more realistic hateful tweets generation. To overcome the problem of differentiation in sequence generation, a policy gradient is used to train the HateGAN model.

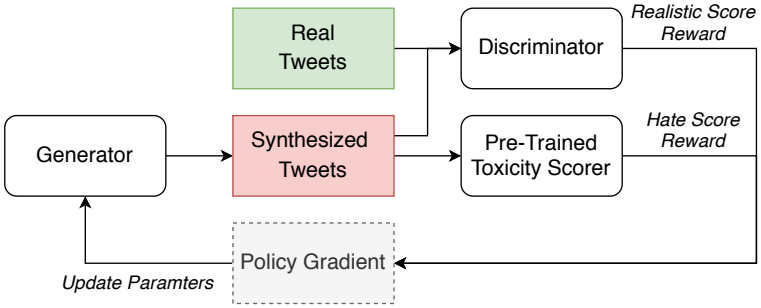

Figure 1: Architecture of the HateGAN model.

3.1 Pre-Training Toxicity Scorer

The toxicity scorer plays a vital role in the HateGAN model as it outputs a *hate score* reward to guide the generator to produce more hateful tweets. Specifically, the toxicity scorer is pre-trained as a multi-label classification model, as shown in Figure 2. The model consists of a word embedding layer, LSTM, and fully connected layer. For word embeddings, we trained a Word2Vec model over tweets in scratched from 17 March 2020 to 28 April 2020 related to the COVID-19 pandemic. The LSTM layer is made up of two stacked LSTM. Max and average pooling operations are applied to all hidden states of the second LSTM. The two vectors are connected to a fully connected layer to generate the final vector for multi-label classification. Binary cross-entropy is used as the loss function. We pre-trained the classification model using the *Toxic Comment Classification Challenge* dataset from Kaggle[2]. The dataset contains

[2]https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

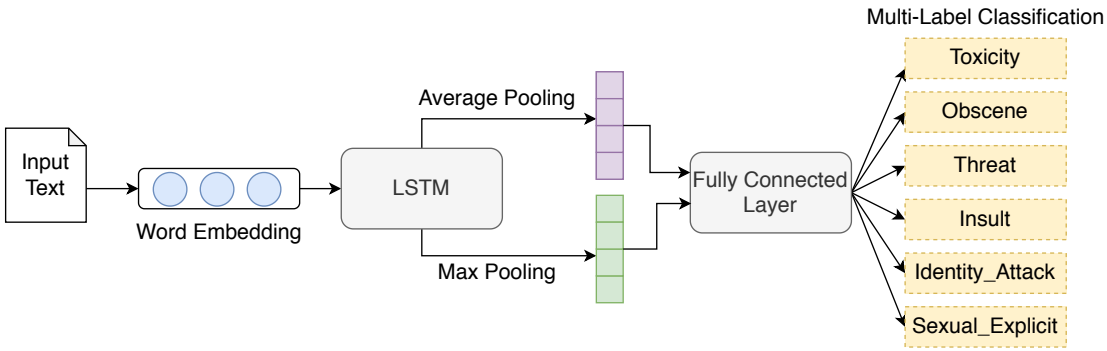

Figure 2: Toxicity scorer pre-trained as a multi-label classification model

about 160,000 comments, and each comment is labeled in six polarities: *toxicity*, *obscene*, *threat*, *insult*, *identity_attack*, and *sexual_explicit*.

Although the toxicity scorer is able to score a given text on the six polarities (i.e., labels in the classification model), not all the polarities are relevant in guiding the generation of hateful tweets. ElSherief et al. (2018) exploited the *toxicity* and *attack_on_commenter* models provided by the *Perspective* API[3] to evaluate and annotate the hatefulness in text. The underlying intuition is that hate speech, by the definition of (Cambridge,), hate speech should both contain hate or violent expression and target groups or individuals. We adopt a similar approach where we consider a given text's *toxicity* and *identity_attack* polarities scores learned by our toxicity scorer. The two scores will be used as hate score rewards to encourage the generator to synthesize more hateful tweets.

3.2 Sequence Generative Model with Policy Gradient

Our HateGAN model adopts an Adversarial-based learning framework where a *Generator* and a *Discriminator* are trained iteratively for tweet generation. The intuition is that the generator aims to generate synthetic data that the discriminator cannot distinguish from real data. However, it is challenging to generate sequence data, such as natural language sentences. For sequence generation, the model gives discrete output, making it hard for gradients to backpropagate. Every token also matters in generating a sequence, and it is meaningless to feed only partially generated sequences to the discriminator. The HateGAN model adopted the sequence generation framework proposed in (Yu et al., 2017), where reinforcement learning and Monte Carlo (MC) search to address the above challenges.

Specifically, we define the sequence generation problem as follows: the generator G_α is required to give an output $W = (w_1, w_2, \cdots, w_n)$, where n is the length of the sequence and α denotes parameters in the generator. When generating for the t-th word, we can regard it as selecting an action from the action space V given the previous state $w_1, w_2, \cdots, w_{t-1}$. The action space V is the vocabulary. After action selection, our scoring module will provide an action reward regarding two aspects: the realisticness and hateful attributes of the generation. Given a synthesized sequence from generator, the scoring module will provide a reward as $Y \in R^n$, elements of which corresponds rewards for each position of the sequence. As discussed earlier, it is ineffective to feed an incomplete sentence into the scoring module. Therefore, when computing reward at time step t, we regard $w_1, \cdots, w_{t-1}$ as state and use Monte Carlo search to finish the rest of generation. The expected reward from a sentence is the action-value for selecting w_t, and it is computed as follow:

$$r(state = (w_1, \cdots, w_{t-1}), action = w_t) = \frac{1}{N}\sum_{i=1}^{N}(S(x_i)) \quad (1)$$

Where S to denote our scoring module, N is the number of MC search and x_i is the *i*-th MC result based on the current state: $w_1, w_2, \cdots, w_{t-1}$. The average rewards from these N sentences will be utilized as the reward for the action reward at time step t. We also noted that for the final word prediction, the MC search is not applied. Instead, we feed the whole generated sentence directly to the scoring module.

[3]https://www.perspectiveapi.com/

Table 1: Statistic information about datasets in experiments

Dataset	#tweets	Classes (#tweets)
WZ-LS	13,202	hate (3,435), non-hate (9,767)
DT	24,783	hate (1,430), offensive (19,190), neither (4,163)
FOUNTA	89,990	normal (53,011), abusive (19,232), spam (13,840), hate (3,907)
HateLingo	11,763	hate (11,763)

Finally, we can obtain a reward vector for sequence: $Y \in R^n$. Thus, for training, the goal is to maximize the expected reward by optimizing parameters in the generator. Specifically, we define the loss as the negative expected reward:

$$\begin{aligned} Loss(\alpha) &= -\sum_{t=1}^{n} E_{[w_{1:t-1}]\sim G_\alpha}[E_{w_t\sim G_\alpha}[r(w_t)]] \\ &\approx -\sum_{t=1}^{n}\sum_{w_t\in V}[G_\alpha(w_t|w_{1:t-1})\frac{1}{N}\sum_{i=1}^{N}S(x_i^t)] \end{aligned} \tag{2}$$

where, S is the scoring module, x_i^t is the i-th search result by Monte Carlo given generated t tokens. Similar to (Yu et al., 2017), we apply policy gradient to optimize the loss in Equation 2.

The realistic and hate score rewards in our model are realized by incorporating a discriminator and toxicity scorer, respectively. The discriminator is a binary classifier trained to evaluate the realisticness of the generated sentence. Parameters in the discriminator are optimized in order to distinguish from the generated tweets from the real ones. The more likely a tweet is perceived to be real, the discriminator will assign a score closer to the max score of 1. The parameter optimization in the discriminator is computed as follows:

$$V(G_\alpha, D_\beta) = E_{x\sim P_{data}}[log(D_\beta(x))] + E_{x\sim P_{G_\alpha}}[log(1 - D_\beta(x))] \tag{3}$$

$$D_\beta^* = \arg\max_\beta V(G_\alpha, D_\beta) \tag{4}$$

where G_α and D_β denotes the generator and discriminator respectively, x is the tokens of sentences sending to the discriminator, β corresponds to the parameters in the discriminator. As the toxicity scorer is pre-trained, its parameters are fixed during the training of HateGAN model. As highlighted in earlier section, we only consider the synthesized text's *toxicity* and *identity_attack* polarities scores for computing the hate score reward. The final combined reward is calculated as follows:

$$r(x) = Discriminator(x) + \sigma ToxcityScorer(x) \tag{5}$$

where r is a numeric value, x is the input sentence, and σ is a hyper-parameter, which balances the contribution from realisticness and hateful attributes for the reward.

4 Experiment

In this section, we describe the experiments conducted to evaluate the effectiveness of HateGAN in augmenting publicly available hate speech datasets and improving the performance of existing hate speech detection models. Specifically, we evaluate the HateGAN's ability to improve the hate speech detection classification performance by augmenting the datasets used in training the classifiers. We also perform analysis to understand the HateGAN's effects on hate speech detection performance by varying the amount of data augmentation. Finally, we demonstrate the quality of HateGAN's generated hateful tweets via case studies.

4.1 Datasets

Four publicly available datasets are used in our study: **WZ-LS** (Waseem and Hovy, 2016), **DT** (Davidson et al., 2017), **FOUNTA** (Founta et al., 2018), and **HateLingo** (ElSherief et al., 2018). The datasets are utilized for two main purposes: (1) training the generator in HateGAN model, and (2) training the classifiers for the hate speech classification task. Training the HateGAN model with a large dataset helps improve the diversity in text generation. Furthermore, having more observations on hateful tweets may also improve the quality of the generated hate tweets. Therefore, we combine the four datasets to train the HateGAN model. Ideally, we could also train the classifier and evaluate the hate speech classification task on the four datasets. However, HateLingo only contains single class tweets (i.e., hate tweets), and recent studies (Awal et al., 2020) have shown significant annotation inconsistency in the FOUNTA dataset. Thus, we focus our evaluation on the WZ-LS and DT datasets. Table 1 shows the distribution of four publicly available datasets used in our study.

4.2 Baselines

To demonstrate the robustness of HateGAN model, we augment the HateGAN generated tweets to improve the performance of three commonly-used deep learning classifiers in hate speech detection: LSTM (Badjatiya et al., 2017; Agrawal and Awekar, 2018; Gröndahl et al., 2018), CNN, (Badjatiya et al., 2017; Agrawal and Awekar, 2018; Gambäck and Sikdar, 2017) and CNN-LSTM (Zhang et al., 2018; Rizos et al., 2019; Cao et al., 2020). We also include the CNN-LSTM-GenAug and CNN-LSTM-TreshAug models proposed by Rizos et al. (2019) as baselines for the DT datasets as the authors have reported their data augmentation results. The CNN-LSTM-GenAug trains a sequence generation model where an LSTM model is trained to predict the next word based on previous states of the LSTM. We can regard it as the first stage of our generative model, which does not have the process of adversarial training and reinforcement learning. The CNN-LSTM-ThreshAug is a substitution-based augmentation method where top-k most similar words are used to replace each word in the original sequence. There is a third method (i.e., CNN-LSTM-PosAug) proposed in (Rizos et al., 2019), which utilized swapping of word positions in a sentence to perform augmentation. However, we argue that this approach greatly disturbs the grammar of the sentence, and it is detrimental to the fluency of the generated sentence. As such, we did not include the word position swapping method as a baseline. Sampling strategies such as upsampling and downsampling are common ways to improve classification performance in imbalanced datasets (Krawczyk, 2016). The goal of these sampling strategies is to sample the data such that the observations of various classes are balanced during training. Thus, we will also compare the HateGAN data augmentation method with the two common sampling strategies.

4.3 Implementation Details

For all deep learning baseline models, the weights of word embeddings are initialized using Glove embeddings (Pennington et al., 2014), the dimension of which is 300. For the toxic comment detection model, we applied the word2vec embeddings trained over a large corpus of tweets, which also have the dimension of 300. To avoid overfitting, 20% and 50% dropouts are applied to embedding layers and fully connected layers. The length of sentences for generation is set as 20. The number of hidden states of LSTM is 200. CNN has 150 filters for the size of filters ranging from 1 to 3, respectively. For our HateGAN model, we pre-trained the generator and discriminator in advance to make the reinforcement learning converge more easily. ADAM (Kingma and Ba, 2014) optimizer is exploited with a learning rate of 0.0001 to train both the HateGAN and baseline models.

4.4 Experiment Results

Table 2 shows the performance on hate speech detection of baselines before and after adding tweets generated by HateGAN model to WZ-LS and DT dataset. Five-fold cross-validations are performed in all experiments, and the average results are reported. Specifically, we evaluate the models' performance by computing the micro averaging F1, which is the preferred averaging evaluation for datasets with class imbalance. As we are interested in models' ability to detect hate speech, we also reported the precision,

Table 2: Performance of baselines before and after data augmentation on WZ-LS and DT datasets. (*) denotes results reported in its original paper.

Model	Micro-F1	Hate-Prec	Hate-Rec	Hate-F1
WZ-LS Dataset				
LSTM	77.3	66.0	46.6	48.2
LSTM-Upsampling	76.8	58.2	63.4	**56.2**
LSTM-Dowmsampling	38.9	44.4	**70.8**	39.0
LSTM+HateGAN	**78.3**	**68.0**	48.6	51.4
CNN	76.7	65.4	47.8	48.2
CNN-Upsampling	77.0	57.8	60.8	**56.2**
CNN-Downsampling	39.5	43.0	**73.8**	39.6
CNN+HateGAN	**78.4**	**68.0**	49.2	52.4
CNN-LSTM	77.2	64.8	49.2	49.0
CNN-LSTM-Upsampling	76.9	59.6	62.0	**56.0**
CNN-LSTM-Downsampling	38.3	42.6	**72.5**	39.8
CNN-LSTM+HateGAN	**78.2**	**65.2**	52.2	53.2
DT Dataset				
LSTM	89.2	50.6	27.4	34.8
LSTM-Upsampling	88.0	34.2	52.4	41.0
LSTM-Downsampling	81.8	48.8	**67.4**	**56.2**
LSTM+HateGAN	**89.6**	**53.2**	28.6	37.0
CNN	89.0	**53.6**	28.4	35.2
CNN-Upsampling	88.5	36.4	48.4	40.8
CNN-Downsampling	82.9	48.4	**72.8**	**58.0**
CNN+HateGAN	**89.5**	50.8	33.2	39.2
CNN-LSTM	88.7	**51.6**	18.6	25.2
CNN-LSTM-Upsampling	87.4	32.2	52.4	38.8
CNN-LSTM-Downsampling	82.6	49.2	**70.0**	**56.7**
CNN-LSTM-GenAug*	-	-	17.0	-
CNN-LSTM-ThreshAug*	-	-	23.0	-
CNN-LSTM+HateGAN	**89.4**	50.4	31.4	37.2

recall, and F1 for the hate class. From Table 2, we observed that the models trained with augmented tweets from HateGAN model are able to outperform the baselines in Micro-F1 in both WZ-LS and DT datasets. In particular, we observe an average 5% improvement in Hate-F1 across all hate speech detection models. This suggests that the data augmentation provided by HateGAN model is robust to improve hate speech detection performance regardless of the base classifier.

Comparing with the sampling strategies, we observed that the Micro-F1 of the sampling strategies are lower than the original baseline performance. A possible reason for this observation is that balancing the dataset in training results in poorer performance of predicting the dominating class in test stage. We also observe a higher Hate-Recall for the hate class and a much lower Hate-Precision from the sampling strategies, suggesting that the classifier might be trained with biases to predict more tweets as hateful. We further examine the effects of HateGAN on the performance in each class by reporting the confusion matrices in Figure 5. We observed that the HateGAN data augmentation brings remarkable improvements on the performance of hate class with little sacrifice on the performance of other classes. For instance, after data augmenting the DT dataset, we observe an increase of 12.8% in correctly predicted hate class tweets while suffering a small decrease of 1.0% in correctly predicted offensive class tweets.

Hate speech detection in the DT dataset is perceived to be more challenging than the WZ-LS dataset as the DT dataset requires multi-class classification. Furthermore, the tweets from the offensive class are known to be similar to those in the hate class. However, the data augmentation from HateGAN is able to improve hate speech detection performance in both WZ-LS and DT datasets. This suggests the hateful tweets generated by HateGAN are diverse and generic enough to augment the different hate speech datasets. More interesting, we also observe that the CNN-LSTM+HateGAN has outperformed the two data augmentation methods proposed in (Rizos et al., 2019), i.e., CNN-LSTM-GenAug and CNN-LSTM-ThreshAug. As the authors did not share the codes for the data augmentation methods, we reported the results shown in the original paper. As observed in Table 2, CNN-LSTM+HateGAN is able to outperform the two data augmentation methods in Hate-Recall. Notably, the CNN-LSTM+HateGAN improvement over the CNN-LSTM-GenAug model suggests that the HateGAN model is able to generate better quality hateful tweets that can improve hate speech detection.

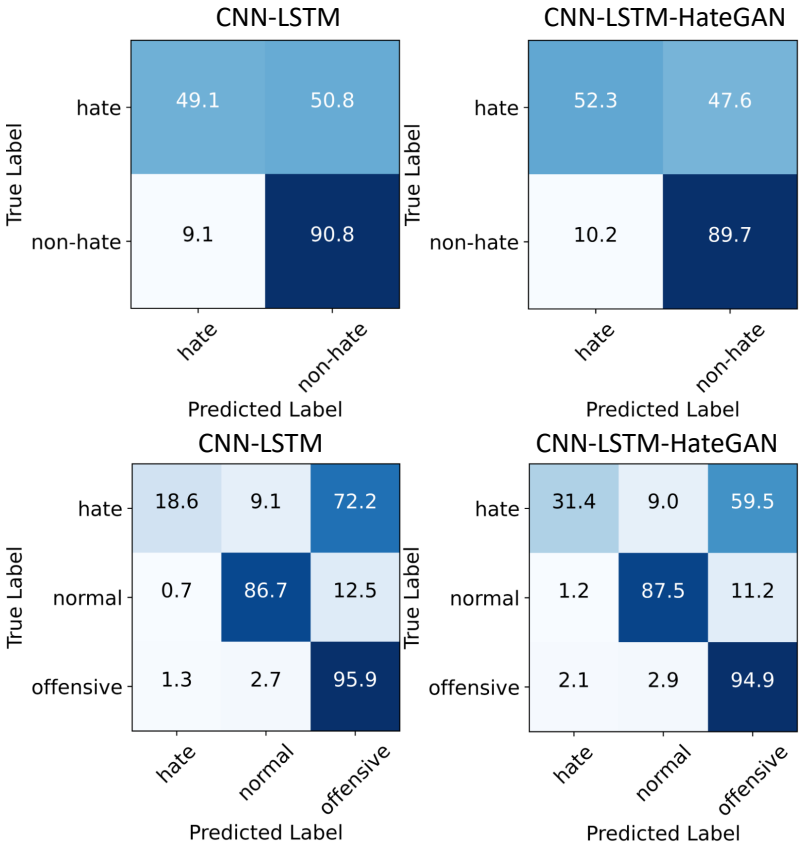

Figure 3: Confusion matrices of CNN-LSTM and CNN-LSTM+HateGAN on WZ-LS (Top) and DT (Bottom) datasets.

4.4.1 Varying Amount of Augmented Tweets

Datasets for hate speech detection are mostly imbalanced. The offensive tweets in DT are about 13 times more than hate ones, while non-hate tweets in WZ-LS are three times more than hate tweets. We postulate that a more balanced dataset will improve the performance of hate speech detection. To explore the relationship between percentages of hate tweets and the performance of hate speech detection, we visualize the Hate-Precision and Hate-F1 scores of the CNN-LSTM+HateGAN model with different percentages of hate tweets augmented in the training dataset in Figure 4. From the figure, we observe an increase for Hate-Precision from the beginning, and a decline follows when we augment more hate tweets. By adding more hate tweets, models are trained over more diversified hateful content, making it easier to detect hate speech. However, as the dataset becomes more balanced, the Hate-Precision drops. A possible reason could be that the classifier tends to predict a given tweet as hateful with increased data augmentation, resulting in lower Hate-Precision scores.

4.5 Case Studies

In this section, we showcase case studies on the hateful tweets generated by the HateGAN model. Specifically, for each generated tweets, we find and compare it with its most similar tweets in the real-world hate speech datasets. The objective is to empirically examine the diversity, coherence, and relevance of the hateful tweets generated by the HateGAN model. To find the corresponding similar tweets in the real-world datasets, we first compute the generated tweets and real-world tweets latent representations using weighted word embeddings. Subsequently, for a given generated tweet, we calculate the cosine similarity between the generated tweet and each tweet in the real-world dataset. Finally, we pair and report the most similar real tweets with the generated tweet.

Table 3 shows the examples of HateGAN generated tweets and their corresponding most similar real tweets in WZ-LS and DT datasets. From the examples, we observe that the generated hateful tweets are not only offensive expressions but also contain hate targets. For instance, the first two tweets contain

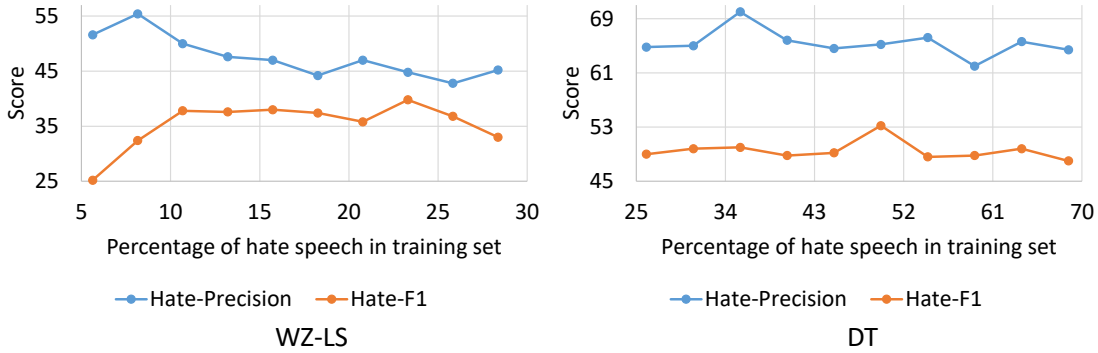

Figure 4: Performance of hate speech detection in various amount of data augmentation.

Table 3: Example of HateGAN generated tweets and their corresponding most similar real tweets in WZ-LS and DT datasets.

Generated Tweets	Most Similar Real Tweets	True Label
you did a statement every night fag*ot you got fag*ot	**[WZ-LS]** RT @Nickodemon1: @YesYoureSexist Shutup if I was sexist I wouldn't be a fag*ot about it, my mums a girl for fu*k sake	hate
do not call me ugly because you fuc*ing fag*ot	**[WZ-LS]** RT @Nickodemon1: @YesYoureSexist Shutup if I was sexist I wouldn't be a fag*ot about it, my mums a girl for fu*k sake	hate
is it a crime we know the true americans are isis just fuc*ing id*ots	**[WZ-LS]** @TVBchelor @NewJerzeyBoy what about the feminazi and the isis member	hate
fu*k off fag*ot as* down	**[DT]** @patrickstpierre You're a fu*king fag*ot	hate
rt pass a female comedian you ugly as* b*tch	**[DT]** RT @Savage_Glam: Idk if @IGGYAZALEA is really a man, but she surely is a trash rapper	normal
you all h*es so d*mn nig*as	**[DT]** RT @LessonswithKate: you got h*es, I got nig*as	offensive

sexism remarks, while the third tweet is targeted at the Americans. Another interesting observation is that our generated tweets are quite different from the most similar real tweets found in the WZ-LS and DT datasets. This suggests that HateGAN is able to generate diverse hateful tweets that are unseen in real-world datasets. However, our last example also supports existing studies, which indicated that it is hard to differentiate offensive tweets from hateful ones (Davidson et al., 2017). In summary, the examples in our case studies show that HateGAN can generate tweets that are diverse, coherent, and relevant to hate speech detection.

5 Conclusion

In this paper, we proposed hateGAN, a deep generative reinforcement learning model, which addressed the challenge of imbalance class by augmenting the dataset with hateful tweets. We conducted extensive experiments to augment two commonly-used hate speech detection datasets with the HateGAN generated tweets. Our experiment results showed that HateGAN improves the detection performance of the hate speech class regardless of the classifiers and datasets used in the detection task. Specifically, we observe an average 5% improvement for the hate class F1 scores across all state-of-the-art hate speech classifiers. We also conducted case studies to empirically examine the HateGAN generated hate speeches and shown that the generated tweets are diverse, coherent, and relevant to hate speech detection. For future work, we aim to explore better methods for generating hateful tweets. For instance,

we will explore other generation models, such as variational auto-encoder, for data augmentation. We will also consider other relevant attributes such as sentiment polarities and topic information, for hate tweets generation. Finally, We will also explore other evaluation methods to measure the quality of the generated hateful content.

References

Sweta Agrawal and Amit Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. In *European Conference on Information Retrieval*, pages 141–153. Springer.

Aymé Arango, Jorge Pérez, and Barbara Poblete. 2019. Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 45–54.

Md Rabiul Awal, Rui Cao, Roy Ka-Wei Lee, and Sandra Mitrović. 2020. On analyzing annotation consistency in online abusive behavior datasets. *arXiv preprint arXiv:2006.13507*.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760.

Bloomberg. 2019. Twitter, facebook join global pledge to fight hate speech online, May.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Cambridge. Hate speech.

Rui Cao, Roy Ka-Wei Lee, and Tuan-Anh Hoang. 2020. Deephate: Hate speech detection via multi-faceted text representations. In *12th ACM Conference on Web Science*, pages 11–20.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. 2017. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on web science conference*, pages 13–22.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 71–80. IEEE.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30.

Bin Dong, Zixing Zhang, Björn Schuller, et al. 2016. Empirical mode decomposition: A data-enrichment perspective on speech emotion recognition. In *Proc. the 6th International Workshop on Emotion and Sentiment Analysis (ESA), satellite of the 10th Language Resources and Evaluation Conference (LREC). Portoroz, Slovenia*, pages 71–75.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth M. Belding. 2018. Hate lingo: A target-based linguistic analysis of hate speech in social media. In *ICWSM*, pages 42–51.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media*.

Antigoni Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2019. A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM Conference on Web Science*, pages 105–114.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90.

Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N Asokan. 2018. All you need is" love" evading hate speech detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, pages 2–12.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Gil Keren, Jun Deng, Jouni Pohjalainen, and Björn W Schuller. 2016. Convolutional neural networks with data augmentation for classifying speakers' native language. In *INTERSPEECH*, pages 2393–2397.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589.

Bartosz Krawczyk. 2016. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 90–98.

Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science*, pages 173–182.

Yashar Mehdad and Joel Tetreault. 2016. Do characters abuse more than words? In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 299–303.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Kunal Relia, Zhengyi Li, Stephanie H Cook, and Rumi Chunara. 2019. Race, ethnicity and national origin-based discrimination in social media and hate crimes across 100 us cities. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 417–427.

Georgios Rizos, Konstantin Hemker, and Björn Schuller. 2019. Augment to prevent: short-text data augmentation in deep learning for hate-speech classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 991–1000.

Justin Salamon and Juan Pablo Bello. 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10.

Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *ICML*.

Times. 2019. Facebook says it's removing more hate speech than ever before. but there's a catch, Nov.

Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. 2017. A bayesian data augmentation approach for learning deep models. In *Advances in neural information processing systems*, pages 2797–2806.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

Matthew Williams. 2019. The connection between online hate speech and real-world hate crime, Oct.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-first AAAI conference on artificial intelligence*.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*, pages 745–760. Springer.