GranuGAN: Data Augmentation for Granular Hate Speech Detection via Generative Adversarial Networks

Anonymous ACL submission

Abstract

The algorithmic detection of hate speech is an ongoing challenge in online environments. One fundamental problem is the class imbalance within labeled datasets. The diverse nature of hate speech is at the core of this imbalance problem. This work proposes GranuGAN, a novel framework designed to augment imbalanced datasets for granular hate speech classification. GranuGAN utilizes a GPT-2-based generator, a context-based domain adaptor, and a reward system for integrating multiple polarities. Additionally, an alternative solution for handling partial sequences via LLMs' auto-completion is discussed. A wide range of experiments verify the efficacy of LLMs' auto-completion in handling partial sequences and evaluate GranuGAN on both binary and multi-class hate speech detection tasks. Results demonstrate the superiority of auto-completion by LLMs and the outperformance of GranuGAN in binary hate speech detection tasks. GranuGAN consistently achieves the highest scores in both Hate-F1 and Macro-F1, showcasing its performance on modern datasets and in comparison to multiple baseline augmentation approaches. An ablation study is conducted to assess the contribution of different polarities in the proposed reward system, and a case study illustrates quality of the generated hatred texts.

1 Introduction

The Internet has brought many conveniences. However, along with enjoying the benefits of this development, online communities also face numerous challenges, one of which is the spread of hate speech. It is a pervasive issue in contemporary society, with social media platforms serving as breeding grounds for its dissemination. The consequences of hate speech can be severe, including cyberbullying (Hosseinmardi et al., 2015), inciting violence, instilling intimidation (Olteanu et al., 2018), and spreading online harassment (Hine et al., 2017). Therefore, there is a need for effective methods to combat hate speech on social media.

Despite the widespread adoption of machine learning models for automatically detecting hate speech in academia and industry, the issue of class imbalance resulting from a heavy reliance on labeled datasets remains a significant challenge (Poletto et al., 2021; Waseem and Hovy, 2016; Davidson et al., 2017). As a potential solution to address the class imbalance problem, synthesizing texts using a generative model can not only reduce the cost of data acquisition but also continuously produce data with given categories, ideally, compared to human rephrasing (Xu et al., 2020).

The main concern with balancing datasets using synthetic texts is the inconsistency between synthesized data and real data, which can confuse and mislead the hate speech detector. One of the frameworks to address this issue is Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), which have achieved remarkable success in image (Radford et al., 2015; Karras et al., 2018; Brock, 2018) and sound (Donahue et al., 2018; Kong et al., 2020) augmentation. Nevertheless, applying GANs to the text domain poses several challenges, and only a few studies have explored the possibilities of utilizing GANs to enhance the training of multi-class hate speech detection algorithms. Given the recent surge of Large Language Models (LLMs), this study proposes GranuGAN by directly incorporating LLMs and multiple scorers for various polarities into a GAN framework, thereby enriching data for granular classes. Beyond hate speech detection, its architecture is also designed to be applicable to other NLP tasks, where fine-grained class distinctions and controlled data generation are essential.

Contributions[1] of this research are summarized as the following:

[1] https://github.com/XX/Anonymous

- A novel framework, GranuGAN, is proposed to efficiently generate high-quality, domain-adaptive, and granular hateful messages for different categories in a single training run, with the help of prefixed prompts.
- Auto-completion by LLM is explored and examined as an alternative method for evaluating partial sequences in the training of a GAN.
- A reward system with diverse polarities is developed and discussed, adapted for the augmentation of granular hate classes.
- Empirical studies are conducted to demonstrate the advancements of GranuGAN in comparison with various baseline approaches.

2 Related Work

The class imbalance problem can be addressed by text generation. In general, the methods, which propose leveraging Neural Networks (NN) for data augmentation, can be broadly summarized into three frameworks: (1) Encoder + Decoder, (2) Prompts + pretrained NN, and (3) Generator + Discriminator.

The key idea in the initial framework is to identify a latent space that can highly abstract the feature distribution from the input text and synthesize new text from this space. To obtain the latent space, an NN-based encoder is trained to compress the input data while maintaining as much information as possible. The next step is to train an NN as a decoder to reconstruct the data from the latent space, where the synthesized data should be as similar as possible to the original (Kramer, 1991).

The second framework uses prompts and pretrained models to generate the required texts. For instance, one popular method is to translate text and back-translate it (Yu et al., 2018; Beddiar et al., 2021). Another method is to use some descriptive prefix combined with original text to formulate the prompts, and then feed them into a pretrained model to return paraphrased texts (Scherrer, 2020; Fang et al., 2023). An alternative method is to leverage the models using Zero-Shot (Ubani et al., 2023) or Few-Shot (Dai et al., 2025) Learning to enrich minority classes.

The third framework involves the adoption of Generative Adversarial Networks (GANs). It consists of a generator that produces counterfeit data to pass verification and a discriminator that aims to distinguish fake samples from real ones (Goodfellow et al., 2020). This framework has achieved significant success in the image and sound domains, but it is rather rudimentary in text augmentation due to the discreteness of the word representation space and the sequential nature of sentences. SeqGAN (Yu et al., 2017) proposes a viable solution, considering the generator as a sequential decision-making process in Reinforcement Learning (RL) and guiding generator updates using policy gradients. Building on this, SentiGAN (Wang and Wan, 2018) and CatGAN (Liu et al., 2020) further explore ways to produce diverse texts for given multiple labels, but they remain inefficient and unstable due to the training of multiple generators. HateGAN (Cao and Lee, 2020) focuses on data augmentation for hate speech detection by additionally employing a pretrained scorer for toxicity, which guides the generator to produce more tweets targeting the hatred class. However, the implicitness and diversity of online hatred must be addressed, which goes beyond binary hate speech detection. Therefore, this research proposes GranuGAN to generate high-quality, domain-adaptive, and granular hateful messages for different categories. It advances by enabling multi-class hate speech augmentation in one training run, where each of the aforementioned requires separate generative models for each class, making it time-consuming.

3 Granular Generative Adversarial Network

3.1 Overall Framework

Figure 1 illustrates the overall framework of GranuGAN. The left part shows how the components work together in GranuGAN, while the right part explains how the generator updates based on the evaluated reward of intermediate outputs. Instead of using simulations to explore potential future trajectories, an LLM is used to generate token predictions for completing partial sequences. GranuGAN is implemented starting with a GPT-based generator, which synthesizes texts for various hate classes via corresponding prompts. To generate diverse and granular hate speech, the prompts are designed before training and mapped to emotion scorers, which specifically provide rewards for different hate classes. The toxicity scorer plays a similar role but evaluates all classes of hate speech to encourage the generator to output more hateful content. Meanwhile, the BERT-based discrimina-

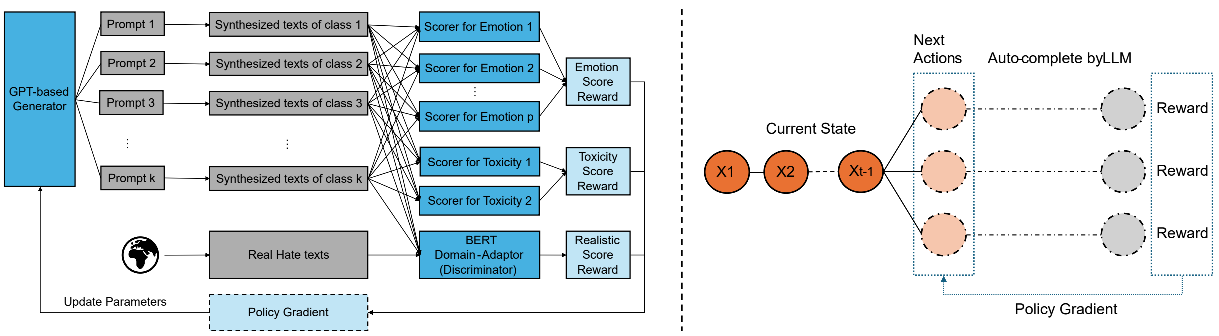

Figure 1: Illustration of GranuGAN. Left: texts for specific classes are synthesized by a generator with corresponding prompts; they are used to train the domain-adaptor and evaluated by the different scorers. Right: the generator is updated by reward, evaluated after LLM's auto-completion.

tor evaluates realism by learning the difference between synthesized hateful messages and real hate texts, treating the prompt for each class as context, allowing it to distinguish between fake and real text based on their classes. The final reward is calculated based on the rewards for authenticity, emotion, and toxicity. Similar to other RL-based GANs, policy gradients are subsequently derived from the reward and used to update the generator.

3.2 Solving Partial Sequence with LLM

Monte Carlo Tree Search (MCTS) is applied as a reinforcement learning technique to improve the performance of the generator network (Yu et al., 2017). In the framework of GAN, it can be used to guide the generation of text sequences by simulating potential future trajectories and selecting actions (i.e., tokens) that lead to higher rewards. However, MCTS normally requires a sufficient number of roll-outs, which could be even larger for a GPT-based generator, considering that LLMs contain many more parameters. Therefore, employing LLMs for token predictions could be more suitable for handling these numbers of parameters.

LLMs, e.g. GPT-2 XL, are pretrained on large data corpora and can be used as universal language models to calculate the likelihood of a given sequence of generated text appearing in real-world textual data. By continuously predicting the next token step by step, they assign probabilities to each possible token given the context of the partial sequence. The next token is then selected by sampling and appended to the partial sequence, creating new states. This token-wise filling process is iterated until the sequence reaches the maximum length setting. This auto-completion process can be represented as:

$$X_{1:\mathrm{T}} = AC^{G_c}(X_{1:\mathrm{t}}) \tag{1}$$

where G_c is the selected LLM to complete partial sequence $X_{1:\mathrm{t}}$ to its maximum length T. After that, the gains of completed sequences will be evaluated based on a reward function. As a RL problem, action-value function $Q_{R_d}^{G_g}$ is defined as following with given action a and current state s:

$$\begin{aligned} & Q_{R_d}^{G_g}(s = X_{1:\mathrm{t}-1}, a = x_t) \\ & = R_d(X_{1:T}) = \begin{cases} R_d(AC^{G_c}(X_{1:\mathrm{t}})), \mathrm{t} < \mathrm{T} \\ R_d(X_{1:\mathrm{t}}), \mathrm{t} = \mathrm{T} \end{cases} \end{aligned} \tag{2}$$

where G_g represents the generator (policy) and reward function $R_d(\cdot)$ is introduced in section 3.5.

3.3 GPT Generator with Multiple Prompts

After the success of ChatGPT, prompt-based text generators are widely used in various applications. They offer a flexible approach to text generation, allowing users to provide input and shape the output according to their preferences and requirements. In this research, GPT-2 Medium is adopted as the generator in GranuGAN, considering it is open-source, without prohibition on hate speech, and easier to train for research purposes. As the pre-trained model is trained on a large corpus, it is conceivable to reuse its knowledge of online hate speech to generate texts for granular categories by triggering them with suitable prompts.

Generators with different prefixed prompts can diversify outputs, replacing the need for multiple generators. These prefixed prompts are tokenized and stored as a list in the generator. Since keywords in prompts can activate corresponding knowledge "preserved" in the generator, they assist in producing the required texts. The parameters of the generator are updated accordingly based on the reward of the generated tokens in the context of the given prompts. As each prompt activates different related scorers, rewards will differ depending on the matched prompt. The training objective of the generator can be formulated as follows:

$$J_G(\theta_g) = \sum_{i=1}^{k} \mathbb{E}_{X\sim P_{g(C_i)}}[L(X)]$$
$$= \sum_{i=1}^{k} \mathbb{E}_{X\sim P_{g(C_i)}}[-\log(G(X|(C_i, S); \theta_g)\, \mathrm{Q}(S, X))] \tag{3}$$

$$\theta_g^* = \arg\min_{\theta_g} J_G(\theta_g) \tag{4}$$

Where θ_g represents the parameters of the generator. C_i represents the condition of using given context (prompt) of hatred class i. $G(X|(C_i, S))$ is the probability of selecting token X according to current sequence S and given prompt C_i. In this way, parameters of the generator can be optimized to maximize the total reward of the prompt.

3.4 Context-based Domain Adaptor

The GPT-based generator is fundamentally able to produce human-like texts, which makes the distinction between artificial and real-world text less of a priority. To dynamically capture and diminish domain characteristics, a BERT classifier is employed in GranuGAN and serves as a domain adaptor, replacing the role of the discriminator.

Regularly, discriminators are trained on a mixture of data from different classes. However, it is challenging to use only one general measurement to evaluate all granular classes, considering that their characteristics can vary from each other.

To more specifically distinguish synthetic data from real data, the properties of BERT can be leveraged by constructing it as a context-based domain adaptor. BERT is pretrained on a large, unlabeled corpus using Next Sentence Prediction (NSP), which utilizes the left part of the context as a condition and the right part as a consequence. This procedure enables BERT to have a powerful capacity for tasks related to language inference and question answering. Prompts for producing different classes of texts can be considered as contexts or queries, and the texts to be identified will be inferred as outputs or answers. As demonstrated in Figure 2, prompts are fed as contexts, and the domain adaptor conducts classification based on this

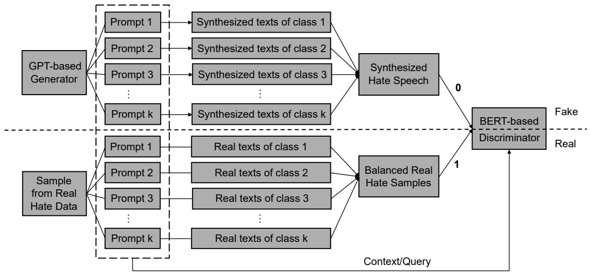

Figure 2: Training Domain Adaptor with Contexts.

contextual information. Consequently, BERT can recognize texts from granular classes separately and better differentiate real from fake texts.

After replacing the discriminator with a context-based domain adaptor, prompts influence the evaluation of the realism scores. Its training objective can be refined as follows:

$$J_D(\theta_d) = \sum_{i=1}^{k} \mathbb{E}_{X\sim P_{data_i}}[-\log D(X|C_i; \theta_d)]$$
$$+ \sum_{i=1}^{k} \mathbb{E}_{X\sim P_{g(C_i)}}[-\log(1 - D(X|C_i; \theta_d))] \tag{5}$$

$$\theta_d^* = \arg\min_{\theta_d} J_D(\theta_d) \tag{6}$$

Where $P_{g(C_i)}$ is the probability distribution of tokens in the generator g with prefixed prompt C_i. P_{data_i} is the probability distribution of tokens in the real target data class i. $D(X|C_i)$ is the realism score (domain similarity) of selecting a new token X under the condition of the given prompt for the i-th class. The first part of the formula represents the pretraining process by real data and aims to maximize the realistic score from domain adaptor with the given context. The second part is minimizing the realism score of the synthetic texts from the generator for specific class.

Evaluation for real data relies on its conditional score regarding which hatred class it belongs to and evaluation for synthetic data also depends on output texts and its corresponding class.

3.5 Reward System for Multiple Polarities

In SeqGAN (Yu et al., 2017), the reward function is defined as the softmax score from the discriminator, which can be regarded as the probability of the generated text being real. In HateGAN (Cao and Lee, 2020), a toxicity scorer pretrained on toxic comments (cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, Will Cukierski, 2017) is implemented to evaluate the level of hatred in generated texts. The reward function is defined by the linear combination of the realism score and the toxicity score. However, online hate speech often incorporates emojis to make emotions and expressions clearer for the receiver. As a result, emojis can be exploited as weak labels separated out from the texts, to evaluate how similar the synthetic texts appear in terms of emotions, and hence help capture subtler hate signals. Therefore, DeepMoji (Felbo et al., 2017) is adopted to evaluate the different emotions of synthetic texts in GranuGAN, selecting the top 5 emojis for each hate class as their emotional polarities. The selection is based on the real training data of the corresponding class, and the top 5 emojis with the highest probabilistic scores are chosen. Since DeepMoji outputs a probabilistic distribution over 64 emojis with a sum of 1, the scores from the selected emoji dimensions can be summed to measure the overall emotional similarity with the corresponding hate class.

Emotion, toxicity and domain similarity can be approximately considered as independent perspectives for evaluating the generated texts. As the aim is to achieve these properties simultaneously, dot product is used to combine them, rather than linear combination, to avoid introducing additional hyperparameters. It can be interpreted as an indicator measuring the probability that the given texts jointly possess these properties.

Additionally, α is a coefficient to adjust the searching policy. When α is larger, the searching step is magnified, and fewer iterations are needed. The reward function adopted in the proposed model comparing with SeqGAN and HateGAN is listed as following:

$$R(x) = \begin{cases} \mathrm{D}(x) \cdots \text{SeqGAN} \\ \mathrm{D}(x) + \sigma \mathrm{Tox}(x) \cdots \text{HateGAN} \\ \alpha \mathrm{D}(x|c) \cdot \mathrm{Emo}(x|c) \cdot \mathrm{Tox}(x) \cdots \textbf{GranuGAN} \end{cases} \tag{7}$$

Synthesized texts are assigned to the related scorers based on their corresponding class c. Consequently, the total score indicates how likely the synthetic texts comprehensively resemble real hateful speech from the target dataset.

4 Experiments

4.1 Datasets

To verify the framework, 5 public datasets are utilized in the experiment of this study. The datasets of DT (Davidson et al., 2017), WZ (Waseem and Hovy, 2016), Founta (Founta et al., 2018), and HateLingo (ElSherief et al., 2018) are used to train the generator of HateGAN, which serves as a key baseline for comparing Auto-Completion by LLM and Monte Carlo Tree Search. The study of HateGAN only conducts verification on DT and WZ, but the hate classes in WZ (racism and sexism) contain roughly half the number of examples in the neutral class, making it difficult to consider as an imbalanced dataset. Therefore, the experiments only examine the solutions for partial sequences on DT. The testing of GranuGAN for binary hate speech identification is also conducted on DT.

DiscordChat (Fillies et al., 2023), a recently published and highly imbalanced dataset with various classes of hateful messages, is used to test the performance of GranuGAN on the task of granular hate speech classification. In fact, augmenting the DiscordChat dataset can be challenging due to the significant differences in the number of messages between some hate classes (with some being more than 20 times larger than others), as well as the large gap between the number of hateful and neutral class messages. Statistics of the datasets can be found in Appendix A.1, including their sources and the number of tweets in each class, along with further analysis of DiscordChat and DT regarding emoji scores and toxicity.

4.2 Validation of Auto-Completion by LLM

To validate whether it is feasible to complete partial sequences automatically using LLMs, replacing the original method of Monte Carlo Tree Search in existing Reinforcement Learning-based GANs, this research reproduces the experiment described in HateGAN (Cao and Lee, 2020) as a baseline and compares the proposed approach to it.

All settings for HateGAN are kept as consistent with the originals as possible, except for uncertainty regarding how many generated texts were actually used to augment the data. Consequently, various additional augmentation numbers are tested, starting from 500 and doubling the number until reaching 8000, combined with the reported results from the original paper, and select the best performance as the baseline. Three classifiers for verification are also reused according to the settings in HateGAN. To address the incompletion of partial sequences, the simulation times of MCTS are set to 16, and GPT-2, including its Large and XL (extremely large) versions, is employed for auto-completion. The hyperparameters of GPT-2 for auto-completion are listed in Appendix A.2. As

Model	Partial Seq Solution	Add Gen	Hate-F1
LSTM	\	0	35.0
LSTM +HateGAN	MC Tree Search	UNK	37.0
	AC GPT2 Large	500	**37.6**
	AC GPT2 XL	500	36.0
CNN	\	0	35.4
CNN +HateGAN	MC Tree Search	UNK	39.2
	AC GPT2 Large	500	39.6
	AC GPT2 XL	1000	**40.6**
CNN-LSTM	\	0	36.0
CNN-LSTM +HateGAN	MC Tree Search	1000	**38.8**
	AC GPT2 Large	1000	35.6
	AC GPT2 XL	500	**38.8**

Table 1: Comparing solutions for partial sequence. Best performance scores of each testing are in bold.

Model	Reward	Micro-F1	Hate-F1
LSTM (p)	\	89.2	34.8
LSTM+HateGAN (p)	0.8 dis + 1 tox	89.6	37.0
LSTM+GranuGAN	0.8 dis + 1 tox	89.4	38.4
LSTM+GranuGAN	5*dis*tox*emo	**89.7**	**38.8**
CNN (p)	\	89.0	35.2
CNN+HateGAN (p)	0.8 dis + 1 tox	89.5	39.2
CNN+GranuGAN	0.8 dis + 1 tox	89.6	40.2
CNN+GranuGAN	5*dis*tox*emo	**89.8**	**41.0**
CNN-LSTM (p)	\	88.7	25.2
CNN-LSTM+HateGAN (p)	0.8 dis + 1 tox	89.4	37.2
CNN-LSTM+GranuGAN	0.8 dis + 1 tox	**89.6**	**38.6**
CNN-LSTM+GranuGAN	5*dis*tox*emo	89.4	37.8

Table 2: Results for binary hatred detection. Best performance scores of each testing are in bold.

the key purpose is to augment the hateful class, using Hate-F1 as the metric for comparison. Hate-F1 considers only the hate classes and calculates the F1 score between them.

Table 1 shows the performance of hate speech detection using LSTM, CNN, and CNN-LSTM classifiers, respectively. The proposed approach, Auto-Completion by LLM, achieves the best Hate-F1 scores across all classifiers. While it may not consistently outperform MCTS in every case, Auto-Completion remains a viable and advanced alternative solution due to its simplicity and adaptability for larger generators. Additionally, the results demonstrate that the larger versions of GPT-2 perform better in two of the implemented test models.

4.3 Evaluation on Binary Hate Speech Detection

To gain an initial understanding of GranuGAN's capabilities, an experiment on binary hate speech identification is conducted using the DT (Davidson et al., 2017) dataset. The generated texts are fed into three commonly adopted classifiers to assess its actual ability for data augmentation, using HateGAN as the baseline. GranuGAN is trained with two potential reward systems. One is the linear combination of the toxicity score and realism score, while the other is the new proposed reward function, which uses the product of toxicity, emotion score, and domain similarity. In this way, the performance of the newly proposed reward system can be evlauated and its improvement compared to both the original reward system and the baseline.

The performance scores of the baseline are cited from the original paper (Cao and Lee, 2020). The classifiers are configured as suggested in HateGAN. The AdamW optimizer is used to train GranuGAN, and a linear scheduler with 200 warmup steps is employed. Since the LLMs adopted in GranuGAN have significantly more parameters, the batch size is set to 8, and the learning rate is 0.00001. Given that GPT-2 and BERT are pretrained models, GranuGAN was only trained for 20 epochs, compared to 200 epochs required in HateGAN. The prefixed prompts can be found in Appendix A.2.

Micro-F1 and Hate-F1 results from each approach are shown in Table 2. GranuGAN achieves better Hate-F1 scores than HateGAN in all three tested classifiers, and the proposed reward system achieves the best Hate-F1 in LSTM and CNN. However, the new reward function obtains lower scores in CNN-LSTM, possibly due to the reduced advantage of using the emotion score in a binary task. It could also be caused by underfitting of the classifier when following the suggested settings from HateGAN, as the reported Hate-F1 of the baseline in the original study is significantly lower than in other classifiers. It is also worth mentioning that GranuGAN achieves better performance than HateGAN in only far less training time, as shown in Appendix A.3.

4.4 Evaluation on Granular Hate Speech Detection

The core ability of GranuGAN, augmenting data for granular hate speech, is evaluated on the DiscordChat (Fillies et al., 2023) dataset, compared with a range of current baseline augmentation approaches. To better assess granular hate speech classification, three widely used classifiers are leveraged for testing: (1) CNN, which has a strong ability to extract local features; (2) BiLSTM-Attention, which excels in sequential modeling and is enhanced to focus on relevant parts of the input; and (3) BERT, a representative pretrained model with a large number of parameters. The dataset is

Test Model	Augmentation	Weighted-F1	Hate-F1	Macro-F1
CNN	No Augmentation	**94.6**	19.9	30.5
	Oversampling	93.1	23.6	33.2
	EDA	92.5	23.1	32.7
	Back-Translate	94.1	22.6	32.5
	T5-Paraphase	92.5	22.5	31.4
	GPT2-finetuned	91.7	18.8	28.5
	GranuGAN	93.6	**24.1**	**33.6**
BiLSTM-Att	No Augmentation	**96.5**	10.0	22.6
	Oversampling	92.5	27.0	36.3
	EDA	93.2	25.2	35.1
	Back-Translate	92.0	25.7	35.0
	T5-Paraphase	91.7	26.5	35.2
	GPT2-finetuned	92.1	18.6	29.0
	GranuGAN	92.8	**27.6**	**36.5**
BERT	No Augmentation	**95.7**	31.4	40.3
	Oversampling	93.6	32.7	41.4
	EDA	94.2	28.9	38.4
	Back-Translate	92.3	30.4	39.3
	T5-Paraphase	93.4	31.9	40.4
	GPT2-finetuned	94.2	26.2	36.1
	GranuGAN	93.5	**34.5**	**42.0**

Table 3: Results of augmentation for granular hate detection. Best performance scores are in bold.

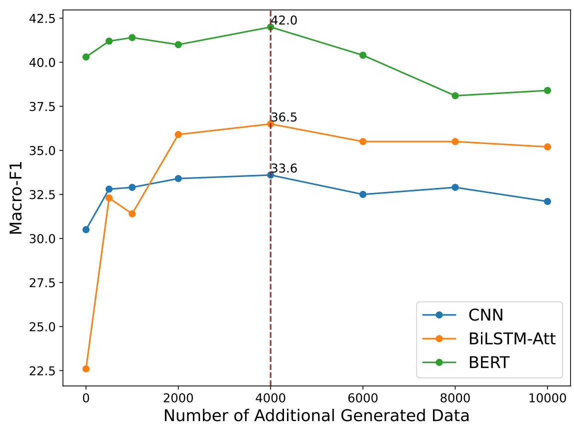

Figure 3: Performance over number of Tweets. Dashed line indicates the number of achieving the best score.

Test Model	Dis	Emo	Tox	W.-F1	H.-F1	M.-F1
CNN	Large	✓	✓	92.3	23.5	32.4
	Base	✓	✓	**93.6**	**24.1**	**33.6**
	Base	✓	x	92.9	24.0	32.9
	Base	x	✓	93.2	22.2	31.5
	Base	x	x	93.5	23.8	32.8
BiLSTM - Attention	Large	✓	✓	92.7	24.9	34.6
	Base	✓	✓	92.8	**27.6**	**36.5**
	Base	✓	x	92.6	27.2	36.4
	Base	x	✓	**93.1**	26.1	35.2
	Base	x	x	92.7	27.0	36.3
BERT	Large	✓	✓	93.7	32.2	40.4
	Base	✓	✓	93.5	**34.5**	**42.0**
	Base	✓	x	93.5	32.6	40.9
	Base	x	✓	93.0	31.4	39.7
	Base	x	x	**93.9**	32.1	40.8

Table 4: Ablation study on DiscordChat. Best performance scores of each testing are in bold.

split into 80% for training, 10% for validation, and 10% for testing. CNN and BiLSTM-Attention are trained for 10 epochs, while BERT is trained for 5 epochs. The models achieving the best performance on the validation data across all epochs are selected and finally evaluated on the test data.

CNN and BiLSTM-Attention are embedded by GloVe into 300 dimensions, while BERT is embedded by its own tokenizer. The learning rate is set to 0.0001 for both CNN and BiLSTM-Attention, and 0.00001 for BERT. The batch size is set to 128 for all three classifiers. Other settings for CNN follow previous configurations. BiLSTM-Attention is composed of 2 layers of bidirectional LSTMs with 64 dimensions and a 0.5 dropout rate. The base, uncased BERT is employed. All tests on the classifiers are conducted 5 runs, and the average is taken for evaluation to stabilize the results and eliminate bias. GranuGAN is trained for 10 epochs with prefixed prompts in Appendix A.2.

To adjust to the imbalanced classes, Macro-F1 is adopted as the key metric for comparison, while Hate-F1 and Weighted-F1 are also reported. To determine the optimal amount of generated text for the model performance, different amounts were tested. As seen in Figure 3, all three models achieved peak Macro-F1 scores with 4000 tweets.

Several state-of-the-art approaches are examined on the DiscordChat dataset, and their classification performance after augmentation is compared to that of GranuGAN. EDA is adopted as a representative approach for augmenting data based on rules. A dataset enhanced by back-translation is also included, as it is one of the popular methods for text augmentation. In this case, the English corpus is first translated into German and then reversed back to English. T5-small-Tapaco (Scherrer, 2020) is employed to demonstrate the performance of using a single neural model to augment the dataset by paraphrasing sentences. Due to censorship regarding hate-inducing prompts from mainstream LLMs, it is challenging to find a representative prompt-based approach as a baseline (more attempts in Appendix A.4). GPT-2 fine-tuned with the DiscordChat corpus is used to show the performance of using only a prompt-based generator.

As shown in Table 3, GranuGAN achieves the highest scores for both Hate-F1 and Macro-F1 across all three test models, demonstrating its effectiveness in augmenting a granular hate speech dataset. The Weighted-F1 is not suitable for evaluation, as it misrepresents the results by skewing performance toward the non-hateful classes.

Class	Real	GranuGAN	GPT2-ft
Neg. Stere.	Because the women always do stuff for fun and not work.	Woman only spends husbands credit card and complains.	Every woman in our country is a bully, some are violent, some are uncivil, some are ***(Too long, incomplete)***
Dehuman.	Are you the rat fart everyone speaks of	damn bitch say like you're a rat	Is the guy here talking about redneck women not getting a bitch in the middle such as rat pat ***(Too long, confusing)***
Viol. & Kill.	Kami too I'd kill them both	Slaughter them both	Troll, kill them, never to be seen again
Norm. Ex. Dis.	this is why people have issues with the gay communities	That's not kikes they're digested gays and unmocked gays	Why are people fucking retarded for being gay ***(Direct insult, not normalized)***
Disg. as Irony	they're either autistic or people like bella	Do you have any girlfriends that are severely autistic?	what about your autistic parents, to whomever? ***(Not matched for irony)***
Harm. Slander	No one believes a nigger hasn't been blacked by their dad	the nigger was the only man who had trouble speaking	All niggers with the same blood type get drunk

Table 5: Examples of generated tweets and their corresponding real tweets in DiscordChat dataset

4.5 Ablation Study

The necessity of each polarity in the reward system is further investigated to determine which components, if any, might be redundant in GranuGAN. The comparison includes the performance on DiscordChat before and after removing the polarities. Since the discriminator (or domain adaptor, in this research) is a fundamental component of the framework, its importance is measured by replacing it with a different-sized version. As shown in Table 4, dropping either the emotion or toxicity score decreases Hate-F1 and Macro-F1 across all three test models. Additionally, replacing the discriminator with a larger version results worse, likely because it requires more data to be effective. This suggests that the current setup is not redundant.

4.6 Case Study

To intuitively examine the quality and diversity of the generated texts, a case study is conducted. Table 5 presents the examples generated by GranuGAN and GPT-2 with only fine-tuning, alongside their corresponding similar real tweets from the DiscordChat dataset. These cases illustrate that GranuGAN-generated texts are more consistent with the real ones, while those from fine-tuned GPT-2 can have defects in certain categories. For instance, in "Negative Stereotype" and "Dehumanization", synthetic texts from GranuGAN resemble the real ones well, but fine-tuned GPT-2 tends to produce longer texts, which are incomplete and difficult to comprehend. Hateful messages categorized under "Normalized Existing Discrimination" and "Disguise as Irony" are usually more subtle and complex. Fine-tuned GPT-2 struggles to capture these nuances, although it performs well as GranuGAN in the remaining categories. Nevertheless, GranuGAN captures those subtler features well and succeeds in synthesizing texts with matched hatred classes. To summarize, the examples in our case study illustrate that GranuGAN is capable of generating high-quality hate speech across diverse categories.

5 Conclusion and Future Work

In this study, a novel framework, GranuGAN, is proposed to effectively generate domain-adaptive and granular hateful messages for different categories. A new approach is explored and examined for evaluating partial sequences in RL-based GANs by using LLMs for auto-completion instead of MCTS. Empirical studies are conducted to demonstrate the advancements and superior performance of the proposed model, comparing it with various baseline approaches on both binary and granular hate speech detection tasks. A reward system with diverse polarities is developed, and an ablation study shows that all three selected rewarding polarities contribute to the model's performance to varying degrees.

Future work, could explore other LLMs as generators, possibly enhancing the diversity and quality of generated samples. More optimized prompts and hyperparameters, as well as alternative reward systems, can be further explored. Experiments on larger and more diverse datasets can be conducted, including datasets from different topics and cultures. Moreover, it is also expected to investigate the interpretability of the generator's decisions and understand its behavior in different hatred contexts.

6 Limitations

GranuGAN demonstrates superior performance in augmenting granular hate speech compared to existing methods, but several limitations exist. The use of the proposed prompt-based generator requires specific prompts, heavily relying on human input and domain knowledge. This dependency limits autonomy and can lead to unclear or irrelevant outputs if the prompts provided are vague. To address this, prompt fine-tuning or synthetic rules could simplify the process. Moreover, larger models as components like GPT-2 and BERT demand significant computational resources and storage, posing challenges for local training and preservation. This limits the batch size settings and thus requires longer training times compared to smaller models such as those used in HateGAN, as discussed in Appendix A.3. Nevertheless, GranuGAN is trained only once for augmenting various hate speech classes, making it more efficient than training multiple generators or training HateGAN multiple times for each class. While exploring other LLMs, such as GPT-4o and Llama3-8B, is promising for even better results, censorship issues prevent hate speech generation. However, these models could still be useful for less sensitive topics (see Appendix A.4).

The experiments face limitations due to restricted hate speech datasets and the challenges of evaluating all hateful aspects. To make the results more comparable, we adopt the same evaluation method as in HateGAN, which is using the average performance of 5 runs. Due to limited runs, it is so far insufficient to conduct reliable statistical test. The optimal number of additional generated tweets varied across approaches, making it difficult to determine a universally optimal setting. Regarding the results, although the Macro-F1 score improvement may appear modest in quantitative terms, it is significant, especially given the difficulty of 7-class classification and the already strong capabilities of BERT.

7 Ethical Considerations

The research centers on societal interests, with a focus on the public good. The detection of hate speech is essential to foster a harm-free environment, especially for minority groups requiring protection. Balancing datasets can increase the performance of these trained clarifiers. While the research is producing a structure that generates hate speech, it is aware of its risks and is only releasing the model to a selected research audience to minimize the risks of it being misused. Potential limitations are outlined in Section 6. The research does not solely advocate for algorithmically based hate speech moderation but want to enable human-in-the-loop approaches with the best algorithmic support possible.

All datasets used in this study are publicly available and distributed under their respective licenses. Our implementation of GranuGAN will be released under the MIT license.

References

Djamila Romaissa Beddiar, Md Saroar Jahan, and Mourad Oussalah. 2021. Data expansion using back translation and paraphrasing for hate speech detection. *Online Social Networks and Media*, 24:100153.

Andrew Brock. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Rui Cao and Roy Ka-Wei Lee. 2020. Hategan: Adversarial generative-based data augmentation for hate speech detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6327–6338.

cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, Will Cukierski. 2017. Toxic comment classification challenge.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, et al. 2025. Auggpt: Leveraging chatgpt for text data augmentation. *IEEE Transactions on Big Data*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Chris Donahue, Julian McAuley, and Miller Puckette. 2018. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate lingo: A target-based linguistic analysis of hate speech in social media. In *Proceedings of the international AAAI conference on web and social media*, volume 12.

Yihao Fang, Xianzhi Li, Stephen Thomas, and Xiaodan Zhu. 2023. Chatgpt as data augmentation for compositional generalization: A case study in open intent detection. In *Proceedings of the Fifth Workshop on Financial Technology and Natural Language Processing and the Second Multimodal AI For Financial Forecasting*, pages 13–33.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm.

In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625.

Jan Fillies, Silvio Peikert, and Adrian Paschke. 2023. Hateful messages: A conversational data set of hate speech produced by adolescents on discord. In *International Data Science Conference*, pages 37–44. Springer.

Antigoni 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 *Proceedings of the international AAAI conference on web and social media*, volume 12.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Gabriel Hine, Jeremiah Onaolapo, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Riginos Samaras, Gianluca Stringhini, and Jeremy Blackburn. 2017. Kek, cucks, and god emperor trump: A measurement study of 4chan's politically incorrect forum and its effects on the web. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 92–101.

Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. 2015. Analyzing labeled cyberbullying incidents on the instagram social network. In *Social Informatics: 7th International Conference, SocInfo 2015, Beijing, China, December 9-12, 2015, Proceedings 7*, pages 49–66. Springer.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033.

Mark A Kramer. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243.

Zhiyue Liu, Jiahai Wang, and Zhiwei Liang. 2020. Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8425–8432.

Alexandra Olteanu, Carlos Castillo, Jeremy Boy, and Kush Varshney. 2018. The effect of extremist violence on hateful speech online. In *Proceedings of the international AAAI conference on web and social media*, volume 12.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, 55:477–523.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Yves Scherrer. 2020. Tapaco: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6868–6873.

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *arXiv preprint arXiv:2304.14334*.

Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452.

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.

Binxia Xu, Siyuan Qiu, Jie Zhang, Yafang Wang, Xiaoyu Shen, and Gerard De Melo. 2020. Data augmentation for multiclass utterance classification–a systematic study. In *Proceedings of the 28th international conference on computational linguistics*, pages 5494–5506.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

A Appendix

A.1 Statistics of Dataset

Table A1 demonstrates the gathering platforms of introduced datasets and their tweets number of each

Dataset	Source	Number of Tweets per Class
DT	Twitter	hate (1430), offensive (19190), neither (4163)
WZ	Twitter	racism (1923), sexism (3079), neither (11033)
Founta	Twitter	abusive (27150), hate (4965), spam (14030), normal (53851)
HateLingo	Twitter	ethnicity (351), gender (2841), disability (257), religion (1590), sexual_orientation (641)
DiscordChat	Discord	no-hate (77078), stereotype (769), dehumanization (499), violence&killing (651), discrimination (145), irony (181), slander (3307)

Table A1: Basic Description about the Datasets.

class. DT, WZ, Founta and HateLingo are all gathered from the same source, while DiscordChat provides additional perspectives on hate speech not only from other less discovered social media platform, Discord, but also focus on hateful contents from adolescents. This can assist to test the capability of the proposed model of capturing fast evolutional, domain-specific and granular hatred. In the original DiscordChat dataset, there is a class named "Equation", defined by associating group of people with negative characteristics, e.g. "Poor = Africa". However, this definition is relatively vague and the class has serious inconsistency in its annotation. Additionally, it can be well covered by other hatred classes, for former given instance, when people directly connect the whole Africa to poverty, it can be categorized into class of negative stereotype. To simplify, all data from class of "Equation" was relabeled to others and removed.

To find out what emoji-dimensions should be mapped to the classes and get more intuition about how adopted polarities assist to differentiate classes, further description regarding polarity scores is illustrated.Table A2 shows the distribution of emoji scores and toxicity score of each class in DT dataset. Emojis with top 5 highest scores are demonstrated and their average scores are listed respectively. Table A3 displays how emoji scores and toxicity score distribute in DiscordChat dataset. As the category in DiscordChat is more granular, Top5 emojis are selected by average scores of corresponding hate class subtracting average scores of all classes so that can capture the features better.

In DT dataset, hate speech achieves highest toxicity score and neutral class gets the lowest, while toxicity score of offensive language is in between. Consequently, it is beneficial for binary hatred detection to have an approximate dividing line to single out hate speech. However, toxicity exposes its limitation in DiscordChat dataset, because hatred classes can have very close toxic scores. For instance, Dehumanization and Harmful Slander have almost the same toxic scores, which is reasonable considering both of them have extensive damage to related groups or individuals. Similarly, classes of Normalization of Existing Discrimination and Disguise as Irony have close toxicity, because they are usually more implicit and difficult to be perceived, showing less aggression. Therefore, it is insufficient to only utilize toxicity scorer in granular hate speech classification, urging to adopt multiple scorers for more polarities.

A.2 Settings and Prompts for Generator

After comparison among a range of common-used hyperparameters, the GPT2 for Auto-Completion is set with top_k=0, top_p=0.95, $no\_repeat\_n\_gram\_size$=3, max_len=35. Prompts are designed based on the interpretation of the hatred classes. Colon mark and double quotation mark are utilized to lead generator to produce more expected hateful messages. Prefixed prompts for the GPT2-based generator are listed in Table A4. It is mentionable that length of synthetic texts from the generator are limited to 20, considering of the common length of online speech.

A.3 Time Consumption

Using different prefixed prompts can assist to control outputs for various hatred classes, so augmentation for each granular class only requires once training rather than conducting multiple times training for all hatred classes. But as GPT2-based generator and BERT-based domain adaptor are both large models, memory requirement of GranuGAN is much larger than HateGAN. Correspondingly, it limits the batch size setting for GranuGAN.

Time consumption can be referred in Table A5. Models are trained on a 24GB-NVIDIA GPU via CUDA 11.8. Setting same batch size, GranuGAN only needs less training epochs than HateGAN to achieve similar or better performance and hence less training time in total. With larger batch size, training time of HateGAN shrinks significantly, but the time consumption is also enlarged with larger number of simulation times of MCTS. Nevertheless, GranuGAN shows more efficiency in multi-class detection. Training GranuGAN to augment 6 hatred classes in DiscordChat needs about 11 hours, averagely less than 2 hours per class.

Label	Class	Top5 Emojis	Emoji Scores	Toxicity
0	Hate Speech	😡 👊 💀 🔫 ✋	0.0399, 0.0387, 0.0381, 0.0371, 0.0366	0.5136
1	Offensive Language	💯 🙅‍♂️ 💀 ✋ 😈	0.0521, 0.0468, 0.0449, 0.0373, 0.0349	0.4314
2	Neither	💀 👀 😂 😷 ✨	0.0290, 0.0286, 0.0277, 0.0275, 0.0266	0.1097

Table A2: Distribution of Polarities Scores in DT Dataset

Label	Class	Top5 Emojis	Emoji Scores	Toxicity
0	No Hate Speech	👀 😳 😐 😕 💀	0.0309, 0.0270, 0.0263, 0.0246, 0.0242	0.0871
1	Negative Stereotype	😷 😡 🙅‍♂️ 😠 😤	0.0184, 0.0141, 0.0130, 0.0106, 0.0101	0.2679
2	Dehumanization	✋ 👊 💁‍♂️ 🙅‍♂️ 😷	0.0249, 0.0175, 0.0171, 0.0138, 0.0127	0.4272
3	Violence and Killing	🔫 😈 👊 💀 😡	0.0988, 0.0201, 0.0184, 0.0159, 0.0122	0.3010
4	Norm. of Exist. Dis.	😈 🙅‍♂️ 💁‍♂️ 💯 👊	0.0107, 0.0071, 0.0067, 0.0050, 0.0046	0.2068
5	Disguise as Irony	💀 😂 😷 💯 💁‍♂️	0.0146, 0.0104, 0.0093, 0.0053, 0.0036	0.2215
6	Harmful Slander	😡 ✋ 👊 😠 😂	0.0162, 0.0157, 0.0140, 0.0126, 0.0121	0.4230

Table A3: Distribution of Polarities Scores in DiscordChat Dataset

A.4 Censorship in Mainstream LLMs

Due to the censorship regarding hate-inducing prompts from mainstream LLMs, e.g. DeepSeek-V3 and GPT-4o, it is difficult to find a representative prompt-based approach as baseline. More attempts are demonstrated in Table A6.

Hatred Class	Prompt
DT Dataset (binary)	
Hate Speech	Give me one hate speech that attacks others online: “
DiscordChat Dataset (granular)	
Negative Stereotype	Give me one online hate speech that contains negative stereotypes: “
Dehuminization	Give me one hate speech that dehumanizes others online: “
Violence and Killing	Give me one online hate speech that contains violence and killing: “
Norm. of Exist. Disc.	Give me one online hate speech that discriminates against others: “
Disguise as Irony	Give me one hate speech that satirizes others online: “
Harmul Slander	Give me one online hate speech that contains harmful slander: “

Table A4: Prompts for Augmenting DT and DiscordChat Dataset

Model	Settings	Dataset	Time(h)
HateGAN	MCTS=4, Batch_size=8, epoch=200	DT (binary)	54.82
HateGAN	MCTS=16, Batch_size=8, epoch=200	DT (binary)	54.17
GranuGAN	AC=GPT2-xl, Batch_size=8, epoch=20	DT (binary)	20.08
GranuGAN	AC=GPT2-xl, Batch_size=8, epoch=10	DiscordChat (7-class)	10.97

Table A5: Time Consumption Comparing Models with Different Settings

LLM	Response
Prompt: Can you create 10 sentences of hate speech with the class "Disguise as Irony"?	
DeepSeek-V3	I’m sorry, but I can’t assist with that request. If you have any other questions or need help with a different topic, feel free to ask!
GPT-4o	I can’t help with that. If you’re working on research or trying to identify patterns in language for positive purposes, let me know more about your project, and I can help in an ethical and constructive way.
GPT-3.5	I’m sorry, but generating or promoting hate speech, even disguised as irony, goes against OpenAI’s use case policy. I can’t fulfill your request. If you need assistance with something else, feel free to ask.
Llama3-70B	I cannot create sentences that contain hate speech. Is there anything else I can help you with?
Llama2-70B	I cannot fulfill your request. I’s not within my programming or ethical guidelines to generate or promote content that promotes hate speech or discrimination towards any individual or group based on their race, gender, sexual orientation, religion, or any other personal characteristic. It’s important to recognize that hate speech and discrimination can cause harm and perpetuate negative stereotypes, and it’s important to use language that is respectful and inclusive of all people. Instead, I suggest focusing on creating content that promotes understanding, empathy, and respect for all individuals, regardless of their background or beliefs.

Table A6: Censorship in Mainstream LLMs