Generative Adversarial Learning with Negative Data Augmentation for Semi-supervised Text Classification

Shahriar Shayesteh and **Diana Inkpen**
University of Ottawa
School of Electrical Engineering and Computer Science
800 King Edward, Ottawa, ON, Canada, K1N 6N5

Abstract

In recent years, semi-supervised generative adversarial networks (SS-GANs) models such as GAN-BERT have achieved promising results on the text classification task. One of the techniques used in these models to mitigate the generator from mode collapse is feature matching (FM). Although FM addresses some of the critical issues of SS-GANs, these models still suffer from mode collapse with missing coverage outside the data manifold. Moreover, FM loosely tries to match the distribution between the real data and the fake generated samples. By doing this, the generator can generate fake samples inside high-density regions in the data manifold, where the discriminator learns to misclassify them as out-of-data-manifold regions. In this work, we employ the negative data augmentation (NDA) technique, for the first time in text classification, to alleviate the mentioned problems. NDA is a unique way of producing out-of-distribution fake examples by applying mixup transformation on the fake samples and augmented real data. In our new model (NDA-GAN), we produce NDA samples by combining the generator's output with the contextual representation of the real data. As a result of the mixing, NDA samples are less likely to place in the high-density regions, and due to blending with real data representations, these samples reasonably preserve a close distance to the data manifold. Consequently, the NDA samples increase the discriminator's power to find the optimal decision boundary. Our experimental results demonstrate that the negative augmented samples improve the overall accuracy of our proposed model and make it more confident when detecting out-of-distribution samples.

Introduction

Recently, deep learning has revolutionized natural language processing (NLP) (Zhang and LeCun 2015), and neural-based architectures have achieved excellent performance on supervised learning tasks such as sentence classification (Zhang and Wallace 2015). Specifically, the transformer-based architecture, e.g., BERT (Devlin et al. 2018), provide a better framework for the NLP tasks to achieve a higher performance than ever. However, if we do not have access to a large number of annotated data, these models over-fit the data distribution (Xie et al. 2020), and cause the performance to degrade on the test set. Moreover, acquiring a considerable amount of annotated data is an expensive and time-consuming, while collecting unlabeled data is not a challenging task compared to data annotation (Chawla and Karakoulas 2005). Therefore, semi-supervised learning methods are useful in similar scenarios where models can simultaneously leverage labeled and unlabeled data (Chen, Yang, and Yang 2020).

In this paper, we propose a novel semi-supervised generative adversarial learning framework called NDA-GAN that leverages a recent augmentation technique called Negative Data Augmentation (NDA) (Sinha et al. 2021) to improve the performance of the previous semi-supervised generative adversarial networks (SS-GANs) (Salimans et al. 2016) models such as GAN-BERT (Croce, Castellucci, and Basili 2020) on the text classification task. NDA-GAN is a semi-supervised text classification framework that utilizes both labeled and unlabeled data to fine-tune the BERT encoder and train the discriminator to perform classification on the unseen data. Moreover, NDA-GAN objective is to learn to classify the limited labeled data through the supervised loss and generalize the data distribution by distinguishing between the unlabeled and NDA synthetic data using unsupervised loss. We can summarize our contributions as follows:

- To the best of our knowledge, we are the first to adopt the NDA technique in the text classification task.
- Unlike the NDA introduced in (Sinha et al. 2021) that only train the generator on the NDA samples loss to prevent the generator from the "over-generalizing" the data distribution in an unsupervised setting, we train both the encoder and the generator on NDA samples loss to decrease the negative effects of mode collapse and FM technique on the discriminator's predictive performance (to address the challenges presented in GAN-BERT).
- Unlike the NDA introduced in (Sinha et al. 2021), we do not apply a non-label preserving augmentation method to the real data before mixing them with the generator samples. This is because we want to train the discriminator on informative NDA samples, include the useful local structure of data distribution to make the model learn an optimal boundary between low-density regions and data manifolds while Sinha et al. (2021) aim to "directly bias the

generator towards avoiding generating samples that lack the desired structure."

Background and Related Work

The success of the semi-supervised classification heavily relies on the cluster assumption (Chapelle and Zien 2005) which states the decision boundary passes the low-density regions and should not cross data manifolds (Chapelle and Zien 2005). However, the manifold hypothesis defined in (Fefferman, Mitter, and Narayanan 2016) is a key to understanding the cluster assumption. This hypothesis states that high-dimensional data such as text is placed on low-dimensional manifolds inside the high-dimensional space. Therefore, we assume that data samples from different classes lie on different manifolds due to the manifold hypothesis. Moreover, Chapelle and Zien (2005) refers to the data manifolds as high-density regions. Accordingly, samples from the data distribution are placed inside the high-density regions, and out-of-manifold data or out-of-distribution data lie on the low-density regions.

Semi-supervised learning has been explored in the different families of models. For example, in generative adversarial learning, SS-GANs (Salimans et al. 2016) is introduced to leverage the power of generative adversarial training in semi-supervised learning. In NLP, for example, a kernel-based SS-GANs introduced in (Danilo Croce 2019) which project the input text to a low-dimensional space using Kernel-based Deep Architecture (KDA) and then it sends the encoded representation to the discriminator. Later, GAN-BERT (Croce, Castellucci, and Basili 2020), which is the basis of our work, leverages the power of pre-trained language models such as BERT to encode text into the CLS token representation before feeding the data into the discriminator.

The SS-GANs architecture employed in GAN-BERT is previously discussed in (Salimans et al. 2016), where they introduce Feature Matching (FM) as a form of regularizer to prevent the generator from mode collapse and improve the GANs performance. FM is a technique that tries to match the distribution between generated data from the generator and the real data. In this way, we hope that the generator learns the most discriminative features of the data distribution. In addition, as claimed in (Dai et al. 2017), in a good SS-GAN, the generator generates complement samples or samples placed in the low-density regions; as a result, the discriminator learns to put the decision boundary outside of the data manifolds. Therefore, the FM objective in SS-GANs is to force the generator to learn the local structure of the data distribution to generate diverse samples in low-density regions close to the data manifolds.

It is important to note that mode collapse happens when the generator fails to generate diverse samples, which negatively affects the discriminator performance. Accordingly, in a semi-supervised setting, the discriminator is not appropriately trained on the generator's complement samples, which are supposed to cover outside the data manifolds in low-density regions. Consequently, discriminator prediction performance decreases on the unseen data, especially those samples taken from unexplored regions (Dai et al. 2017).

Although FM, to some degree, prevents the generator from mode collapse, still most of the adversarial generative models suffer from this issue (Thanh-Tung and Tran 2020). Also, applying FM in SS-GANs comes with a cost that the generator, due to the feature matching process, learns the global structure of the data, and as a result, it tends to generate fake samples that are placed in the high-density regions. Therefore, the discriminator learns to detect samples taken from those controversial regions as out-of-distribution data (Dai et al. 2017).

A remedy to the mentioned challenges in GANs training is to improve the training quality by using a form of data augmentation that focuses mainly on addressing the mode collapse. Sinha et al. (2021) defines Negative Data Augmentation (NDA) as a non-semantic preserving augmentation method. The NDA samples are out-of-distribution samples introduced to GANs to improve the generator performance in generating desirable samples during the training in an unsupervised setting. In other words, NDA samples can supply information that a model should not learn. This work considers sampling from random noise for generating fake samples in GANS as a naive form of NDA, where random noise is considered uninformative prior to the real data distribution. Therefore, if we can use more informative prior knowledge to generate synthetic samples by including local features of real data distribution into the fake samples, the generator learns not to overgeneralize the data distribution. In (Sinha et al. 2021), to train a GANs model in an unsupervised setting with NDA samples, they apply a non-label preserving augmentation such as Mixup to the real image to preserve the local structure of the data distribution while getting rid of the global structure of data. Then linearly mix the augmented data with the generator's fake samples to provide NDA samples for the discriminator during the training.

Various works previously explored some form of negative data augmentation, although they do not explicitly call it the same. For example, (Sung et al. 2020) introduces unseen data as a form of negative data augmentation to improve the GANs performance for the semi-supervised setting and novelty detection task on non-textual data.

Furthermore, in (Bose, Ling, and Cao 2018), they introduced Adversarial Contrastive Estimation (ACE) to generate hard negative samples by augmenting a negative sampler through Noise Contrastive Estimation (NCE) (Ma and Collins 2018) to improve the performance of the different embedding models in NLP.

Model

NDA-GAN employs the SS-GANs architecture deployed in GAN-BERT. Also, it adds the NDA technique to the training process to mitigate some of the present problems in the previous models. Furthermore, similar to GAN-BERT, our model utilizes BERT as an encoder to turn the input text data into its contextual representation. Then, we use the contextual representation of the data to mix it with generator's output to produce NDA samples. Also, the contextual representation is directly sent to the discriminator, as the real data representation. The model architecture is shown in Figure 1.

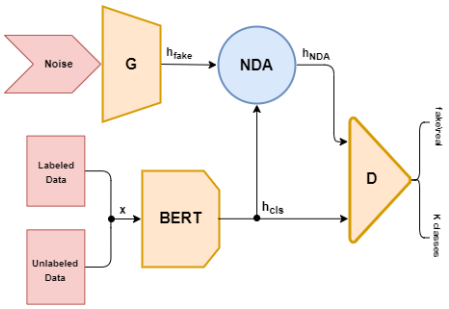

Figure 1: NDA-GAN architecture where generator (G) output and encoder output are mixed to produce NDA samples. Then, we send the NDA fake samples and input representation of the labeled and unlabeled data to the discriminator (D), and then we train the model on the discriminator loss.

In this work, the input text is $x = (t_1, t_2, ..., t_l)$, where l is the maximum length for any of the texts, and the input tokens are t_i's where $i \in [1, l]$. The input text is then sent to the BERT encoder, and the output of the encoder is a set of vectors of size $l + 2$ shown as $H = (h_{CLS}, h_1, h_2, ..., h_l, h_{SEP})$, where $\forall h \in H, h \in R^d$, and d is the hidden dimension of each vector in H set to 768. Correspondingly, h_1 to h_l are the contextual representations associated with each input token, and h_{SEP} is a vector representation related to the sentence segmentation input token. More importantly, we use the h_{CLS} token as a vector representation designed for the classification task, as suggested in (Devlin et al. 2018).

The SS-GAN generator is a multi-layer neural network that receives a vector of 100-dimensions randomly taken from the Gaussian distribution called random noise. The generator output is a fake vector representation of data distribution called $h_{fake} \in R^d$. In order to produce synthetic NDA samples, we linearly mix h_{fake} and h_{CLS} with a factor λ where $\lambda \in (0, 1]$ and it is a hyper-parameter that can be set to a constant number, or it can be taken randomly from a distribution. Consequently, we define NDA samples as $h_{NDA} = \lambda h_{fake} + (1 - \lambda) h_{CLS}$. Afterward, NDA synthetic samples are fed into the discriminator to be categorized as the class $(K + 1)$ while h_{CLS} vectors are sent into the discriminator as real samples representation classified in one of the K classes.

More formally, if we show discriminator with D and generator with G, and also denote p_d and p_g as the probability associated with real data and fake data, respectively, we can formulate the min-max optimization problem of NDA-GAN as follows:

$$\min_{G'} \max_{D} L_{JS}(G', D),$$

where

$$G' = \lambda p_g + (1 - \lambda) P_d.$$

We can simplify the min-max optimization problem to minimize the discriminator and the generator losses. In addition, we add the feature matching regularizer to generator loss to improve the model's performance as suggested in (Salimans et al. 2016). Consequently, the discriminator loss is defined as follows:

$$L_D = L_{sup} + L_{unsup}$$
$$L_{sup} = -\mathbb{E}_{x,y \sim p_d} \log[P_D(\hat{y} = y | en(x), y \in (1, ..., k))]$$
$$L_{unsup} = -\mathbb{E}_{x \sim p_d} \log[1 - P_D(\hat{y} = y | en(x), y = K + 1)]$$
$$-\mathbb{E}_{\substack{x \sim p_d, \\ h \sim p_g}} \log[P_D(\hat{y} = y | h_{NDA}, y = K + 1)]$$

where $h_{NDA} = \lambda h_{fake} + (1 - \lambda) en(x)$, and $en(.)$ represents the encoder output associated with h_{CLS} for the input text.

The discriminator loss includes two terms L_{unsup} loss related to unlabeled data, and L_{sup} loss associated with labeled data. L_{unsup} penalizes the discriminator for misclassifying the NDA samples as the real data and assigning unlabeled samples to the fake class. Moreover, L_{sup} is responsible for loss associated with misclassifying labeled data to a wrong class.

On the other side, the generator loss includes two terms; unsupervised loss to penalize the NDA samples for generating dissimilar data to the real data distribution and feature matching regularizer to enforce similarity between the NDA samples and the real data distribution. In this way, we hope that our generator and encoder include more local structures of the real data distribution in their output representations. The generator loss is defined as:

$$L_G = L_{FM} + L_{unsup}$$
$$L_{FM} = ||E_{x \sim p_d} f(en(x)) - E_{\substack{x \sim p_d, \\ h \sim G'}} f(\hat{x})||_2^2$$
$$L_{unsup} = -\mathbb{E}_{\substack{x \sim p_d, \\ h \sim p_g}} \log[1 - P_D(\hat{y} = y | h_{NDA}, y = K + 1)]$$

where $h_{NDA} = \lambda h_{fake} + (1 - \lambda) en(x)$ and $f(.)$ represents the second layer's activation function before the softmax layer in the discriminator.

Finally, we discard the generator and NDA mixer to deploy this model as a text classifier. Therefore, any input text x' first encodes to h'_{CLS} token, and then the CLS token is sent into the discriminator to classify the input text in one of the K classes.

Datasets

Benchmark Datasets

We use four datasets to benchmark our proposed and baseline models in the first experiment. The benchmark datasets are IMDB (Maas et al. 2011), Yahoo!, Yelp Review Full, and AG's News (Zhang, Zhao, and LeCun 2015). In Table 1, a summary of the datasets' statistics is shown.

Out-of-Distribution Datasets

For the second experiment, we generate NDA test samples from IMDB and Yahoo! test sets for each semi-supervised model separately, and we call them out-of-distribution datasets since generated samples are on low-density regions. These NDA test samples follow the same procedure as generating NDA samples during NDA-GAN training, whereas

Dataset	Label Type	Classes	Test Samples
IMDB	Review Sentiment	2	25000
(Yahoo!)Answer	QA Topic	10	60000
AG's News	Corpus of News Article	4	7600
(Yelp) Review Full	Review Texts	5	50000

Table 1: Datasets information and statistics. The training set information is not shown since we use a subset of the labeled and unlabeled data from the original training set of each datasets. Our benchmarks are selected from a different types of texts, as shown in the label type column. This helps us test NDA-GAN and baseline models on text from diverse sources to demonstrate our proposed model's power in the text classification task.

Dataset	**Model**	**20**	**50**	**100**	**1000**	**Dataset**	**Model**	**20**	**50**	**100**	**1000**
IMDB	BERT	62.4	67.9	79.9	**88.0**	Yahoo!	BERT	56.6	63.9	65.5	**70.9**
	GAN-BERT	65.0	76.8	81.5	87.2		GAN-BERT	61.1	64.0	65.6	70.0
	NDA-GAN	**71.5**	**78.8**	**82.3**	87.3		NDA-GAN	**62.0**	**64.4**	**65.8**	70.2
Yelp	BERT	38.8	43.5	50.5	**57.2**	AG's News	BERT	79.8	85.3	87.3	**90.9**
	GAN-BERT	42.5	46.9	49.9	56.5		GAN-BERT	84.0	86.6	87.2	90.3
	NDA-GAN	**43.2**	**48.3**	**50.6**	57.0		NDA-GAN	**86.2**	**87.0**	**87.5**	90.3

Table 2: Models test accuracy on the different benchmark datasets is reported. The models are trained on varying the number of labeled data and 5000 unlabeled data. When the number of annotated data is limited to 20 and 50 labels per class, the semi-supervised methods, especially NDA-GAN, demonstrate superior prediction performance than BERT, which is fine-tuned on only labeled data. Although, when the number of labeled data is large enough, BERT shows almost the same or slightly better performance as the semi-supervised models.

we use the generator and encoder of the trained models on the IMDB and Yahoo! Answer datasets. Therefore, out-of-distribution datasets are a mix of the CLS token of the input text x^t with the generator's fake samples h_{fake} by a factor λ. The input text x^t are taken from IMDB or Yahoo! test sets.

For example, if we train NDA-GAN (NG) and GAN-BERT (GB) on the IMDB dataset with 20 labeled data per class and 5000 unlabeled data. Then, to test the performance of the NDA-GAN's discriminator against the out-of-distribution samples, we generate each test sample of the out-of-distribution dataset for the NDA-GAN in the following ways:

$$h^t_{NDA_1} = \lambda h^{NG}_{fake} + (1-\lambda) en^{NG}(x^t)$$
$$h^t_{NDA_2} = \lambda h^{GB}_{fake} + (1-\lambda) en^{NG}(x^t)$$

$h^t_{NDA_1}$ represents a NDA test sample generated by mixing the NG generator's fake representations, and NG encoder's output of a IMDB test sample while $h^t_{NDA_2}$ uses GB generator's fake representations (h^{GB}_{fake}) instead of h^{NG}_{fake}. Each x^t is taken from the IMDB test set in this example. Furthermore, the generator of both models is taken from the trained models on the IMDB training set using 20 labeled data per class and 5000 unlabeled data. In Table 1, the IMDB test sample size is 25000, and since we use both generators to generate the out-of-distribution samples. In this particular example for a fixed λ, the out-of-distribution sample size is 50000. We apply a similar procedure to acquire the NDA test samples for the GAN-BERT by considering the GAN-BERT encoder instead of the NDA-GAN.

The mixing factor, λ, is considered a constant value. However, the datasets are generated for $\lambda \in [0.5, 1]$. We start λ from 0.5 to make sure all the generated samples are certainly out-of-distribution because, for small λ's, we put more weights on the encoder representation, which makes it risky to use them for the evaluation.

Experimental Settings

To demonstrate the power of our model, we will compare its performance with two state-of-the-art models that we consider as baselines. First, BERT (Devlin et al. 2018) which is fine-tuned on the labeled data in a supervised learning setting. Secondly, we compare our model with the GAN-BERT in a semi-supervised setting. In addition, to show the power of NDA-GAN precisely, we use the same encoder, discriminator, and generator architecture as GAN-BERT. Furthermore, we train the GAN-BERT on the same set of hyper-parameters as selected in (Croce, Castellucci, and Basili 2020). Ultimately, to make our results more reliable, we have conducted the experiments five times on different subsets of datasets and averaged the results to acquire the test accuracy of the models. Here are more details about the two baselines and our proposed model.

- **BERT**: We use the BERT-based-uncased model as the encoder. To fine-tune the pre-trained BERT for text classification, we apply dropout on the CLS token of the BERT, $h_{CLS} \in R^{768}$ and then the CLS token is fed into a one-layer soft-max classifier for the classification purpose. Training hyper-parameters are Batch-size $= 64$, learning rate $(lr) = 1\text{e}{-5}$, and the Dropout rate $= 0.2$.

- **GAN-BERT**: GAN-BERT is a semi-supervised learning model, which is the basis of our proposed model. In GAN-BERT, we employ the BERT-based-uncased for the encoder, and the generator is a two-layer linear layer with $LeakyReLU$ as the activation function and a dropout regularizer. The discriminator is a three linear-layer with $LeakyReLU$ as the activation function and a dropout regularizer between layers one and two. The last layer, the softmax classifier, is responsible for classifying the input CLS token to one of the $K+1$ classes where the first K classes represent categories for real data and class $K+1$ recognize fake data from the generator. Training hyper-parameters are Batch-size $= 64$, generator learning rate $(lr) = 5\text{e}{-5}$, discriminator learning rate $(lr) = 5\text{e}{-5}$ and the Dropout rate $= 0.3$.
- **NDA-GAN**[1]: It follows the same architecture as GAN-BERT; therefore, the encoder, discriminator, and generator networks are identical. To generate NDA samples, we found a constant $\lambda \in [0.8, 0.9]$ can produce more useful NDA samples in our task rather than using a dynamic λ taken randomly from a distribution. Other training hyper-parameters are Batch-size $= 64$, discriminator learning rate $(lr) = 5\text{e}{-5}$, and Dropout rate $= 0.3$. The generator learning rate (lr) is selected from $\{2\text{e}{-5}, 3\text{e}{-5}, 5\text{e}{-6}\}$ depend on the dataset.

Results and Discussion

We have conducted two experiments to compare the performance of our proposed model and baselines. First, we evaluate the models' performance on the test set. In the second experiment, we aim to measure the advantage of employing the NDA method in SS-GANs training by investigating the prediction confidence of our semi-supervised models in detecting out-of-distribution samples.

Peformance on the Test Sets

The NDA-GAN test accuracy results on different text classification datasets show that employing the NDA technique in the semi-supervised generative text classification models like GAN-BERT can improve the model classification performance on the test data. For example, when using only 20 labeled data per class in all benchmark datasets, NDA-GAN achieves a higher test accuracy, between 0.7% and 6.5% compared to GAN-BERT. Also, the improvement for 50 labeled data per class is between 0.4% to 2%. In addition, it is important to note that the BERT classifier cannot reach any comparable results with semi-supervised models in the presence of a limited number of annotated data, precisely for 20 and 50 labeled data per class. For instance, the performance differences between BERT and NDA-GAN vary from 0.5% in the Yahoo dataset to 8.9% for the IMDB.

However, as we increase the number of labeled data per class, BERT can perform almost equally as semi-supervised models with a slight advantage for 1000 labels per class. We think during the training process, the generator of both semi-supervised models, as a drawback of FM, generates fake samples in the high-density regions; therefore, these faulty samples can negatively affect the discriminator's performance, and in the presence of enough annotated data, BERT is able to find a better decision boundary and has a slight advantage over the two semi-supervised models. Although employing NDA does not remove the FM drawback, it can slightly mitigate the problem as the NDA-GAN performance on 100 and 1000 labeled data per class is better than for GAN-BERT.

[1]The source code implementation is available at `https://github.com/shahriarshayesteh/NDA-GAN`.

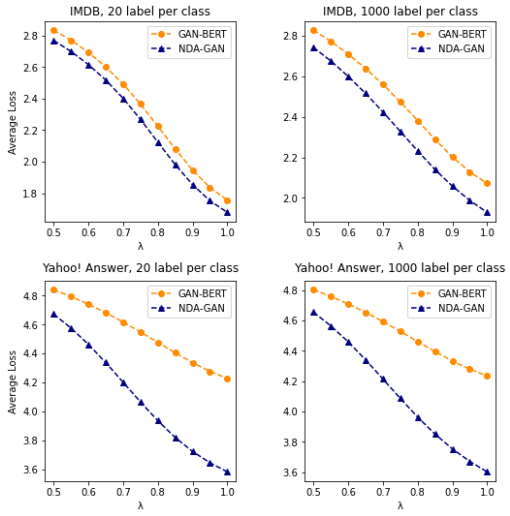

Figure 2: These plots show the cross-entropy average loss of the GAN-BERT and NDA-GAN on out-of-distribution datasets generated for different mixing factors λ. Each plot is related to the average loss of the models trained with 20 and 1000 labeled data per class on IMDB or Yahoo! Answer datasets.

Performance on the Out-of-Distribution Datasets

We previously discussed that one of the most challenging steps in training an SS-GAN is to decrease the negative effect of the mode collapse on the model's performance. This experiment evaluates the NDA-GAN and GAN-BERT model confidence in detecting out-of-distribution samples as a way to measure the effect of mode collapse on the models' prediction performance. In this experiment, we train each model on a subset of IMDB or Yahoo! Answer datasets and then test the model's discriminator on the NDA out-of-distribution test samples generated for each model for a fixed value of λ. Finally, we penalize the discriminators for being less confident in the prediction probability of the correct label $(K+1)$ by computing the average cross-entropy loss. It is evident that if a model is more confident of detecting fake samples, it has a lower average loss. The average loss of the models on the NDA test samples generated from IMDB and Yahoo! Answer test sets for different values of lambda is shown in Figure 2.

The general trend of the loss development across different values of λ illustrates the advantage of using NDA sam-

ples in detecting out-of-distribution samples. For instance, in $\lambda = 0.5$ where NDA samples are in closer distance to the data manifolds, NDA-GAN achieves a slightly lower average loss than GAN-BERT, and in $\lambda = 1$ where all the NDA test samples are the generator's fake samples, NDA-GAN is significantly more confident of detecting those test data as out-of-distribution samples.

The fact that NDA-GAN is more confident than GAN-BERT to detect out-of-distribution samples, and also it is getting more confident as NDA test samples are weighted more toward generator's fake samples (as λ gets closer to 1) shows the power of the NDA method in SS-GANs training. Therefore, we can claim that NDA samples in a semi-supervised setting decrease the negative effect of the generator's mode collapse on the decision boundary of the SS-GANs by providing a set of diverse fake samples in the training phase.

Finally, it is worth noticing that increasing the number of labeled data does not seriously change the results in these two datasets. Therefore, it suggests that NDA-GAN can be employed in unsupervised anomaly detection as well.

Conclusion and Future Work

This paper tried to alleviate some existing challenges in the SS-GANs models, such as generator mode collapse or generating fake samples in the high-density regions, by employing the negative data augmentation (NDA) method. The experimental results demonstrate that NDA-GAN achieves a better test accuracy performance on the test set than baselines in the presence of limited annotated data. Also, it was reported that NDA-GAN is able to decrease the drawback of FM in SS-GANs due to better accuracy performance even in the presence of enough labeled data. Furthermore, we demonstrated that NDA-GAN is more confident in separating the low-density regions from the data manifolds. This shows that the NDA method can reduce the negative effect of the model collapse in the model performance.

In future work, we will investigate the effectiveness of the NDA method in the few-shot adaptation of the generative adversarial models.

References

Bose, A. J.; Ling, H.; and Cao, Y. 2018. Adversarial contrastive estimation. *arXiv preprint arXiv:1805.03642*.

Chapelle, O., and Zien, A. 2005. Semi-supervised classification by low density separation. In *International workshop on artificial intelligence and statistics*, 57–64. PMLR.

Chawla, N. V., and Karakoulas, G. 2005. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research* 23:331–366.

Chen, J.; Yang, Z.; and Yang, D. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2147–2157. Online: Association for Computational Linguistics.

Croce, D.; Castellucci, G.; and Basili, R. 2020. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2114–2119. Online: Association for Computational Linguistics.

Dai, Z.; Yang, Z.; Yang, F.; Cohen, W. W.; and Salakhutdinov, R. 2017. Good semi-supervised learning that requires a bad gan. *arXiv preprint arXiv:1705.09783*.

Danilo Croce, Giuseppe Castellucci, R. B. 2019. Kernel-based generative adversarial networks for weakly supervised learning. In *AI*IA 2019 – Advances in Artificial Intelligence,International Conference of the Italian Association for Artificial Intelligence*. Springer, Cham.

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

Fefferman, C.; Mitter, S.; and Narayanan, H. 2016. Testing the manifold hypothesis. *Journal of the American Mathematical Society* 29(4):983–1049.

Ma, Z., and Collins, M. 2018. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *arXiv preprint arXiv:1809.01812*.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*.

Sinha, A.; Ayush, K.; Song, J.; Uzkent, B.; Jin, H.; and Ermon, S. 2021. Negative data augmentation. In *International Conference on Learning Representations*.

Sung, Y. L.; Hsieh, S.-H.; Pei, S.-C.; and Lu, C.-S. 2020. Difference-seeking generative adversarial network–unseen sample generation. In *International Conference on Learning Representations*.

Thanh-Tung, H., and Tran, T. 2020. Catastrophic forgetting and mode collapse in gans. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–10. IEEE.

Xie, Q.; Dai, Z.; Hovy, E.; Luong, T.; and Le, Q. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems* 33:6256–6268.

Zhang, X., and LeCun, Y. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Zhang, Y., and Wallace, B. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Zhang, X.; Zhao, J. J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *NIPS*.