GAN with BERT – Generative Adversarial Network for Semi-Supervised learning

Jiayue Lin
Virginia Tech
jiayuelin@vt.edu

Ziyi Wei
Virginia Tech
ziyiw@vt.edu

Abstract

This project delves into the domain of Natural Language Processing (NLP) with a focus on semi-supervised learning. It encompasses a review of pertinent literature in this area and provides an introductory overview of GAN (Generative Adversarial Network) alongside our bespoke model, GAN-BERT. The experimental phase involved evaluating the GAN-BERT model on the IMDB and SNLI datasets, utilizing only one percent of labeled data.

For the IMDB sentiment analysis task, the GAN-BERT model demonstrated remarkable performance, achieving 74.9% and 74.6% accuracy on the training and test sets, respectively. Leveraging a semi-supervised learning approach and consistent training parameters with the pre-trained BERT model significantly improved accuracy, highlighting the potential of limited labeled data in enhancing sentiment analysis.

Conversely, in the context of the SNLI dataset for natural language inference, the GAN-BERT model displayed comparatively lower performance, achieving 32.9% and 32.7% accuracy on the training and test sets. Challenges stemming from limited labeled data and the inherent complexities of GANs affected the model's ability to capture intricate patterns, thus impacting its performance.

This study underscores the potential of semi-supervised learning techniques in NLP tasks, especially in sentiment analysis, while highlighting the challenges associated with limited labeled data. Future work involves exploring strategies to increase labeled data proportions, fine-tuning hyperparameters, and extending training duration to enhance GAN-BERT's performance across various NLP tasks.

1 Introduction

Recent years have seen the rapid rise of deep learning approaches in Natural Language Processing (NLP) (Ficler and Goldberg, 2016). Especially, Transformer-based models can parallelize computation without considering the sequential information suitable for large scale datasets, making it popular for NLP tasks. Thus, some other works are used for text classification tasks and get excellent performance. Since the release of Bidirectional Encoder Representations from Transformers (BERT)(Devlin et al., 2019), it has emerged as a significant breakthrough in the field of NLP, rapidly propelling advancements in the domain. BERT, proposed by Google researchers, introduces a pretrained language representation model leveraging a deep bidirectional Transformer architecture, successfully capturing rich semantic information within bidirectional contexts. Most of the adopted benchmarks for BERT are made of thousands of examples. However, in many real scenarios, obtaining high quality annotated data is a time-consuming and costly process.

For the scenario in the text classification, which is the procedure of designating predefined labels for text. It is an essential and significant task in many NLP applications, such as sentiment analysis, topic labeling, question answering and so on(Li et al., 2022). In the era of information explosion, it is time-consuming and challenging to process and classify large amounts of text data manually. Besides, the accuracy of manual text classification can be easily influenced by human factors, such as fatigue and expertise. It is desirable to use machine learning methods to automate the text classification procedure to yield more reliable and less subjective results. Moreover, this can also help enhance information retrieval efficiency and alleviate the problem of information overload by locating the required information.

As a result, many people are turning their attention to semi-supervised learning area to find a viable solution for this problem (He et al., 2018). Semi-supervised learning is the branch of machine

learning concerned with using labelled as well as unlabelled data to perform certain learning tasks. Conceptually situated between supervised and unsupervised learning, it permits harnessing the large amounts of unlabelled data available in many use cases in combination with typically smaller sets of labelled data. In recent years, research in this area has followed the general trends observed in machine learning, with much attention directed at neural network-based models and generative learning.

Over the past two decades, a broad variety of semi-supervised classification algorithms has been proposed (Van Engelen and Hoos, 2020). These methods differ in the semi-supervised learning assumptions they are based on, in how they make use of unlabelled data, and in the way they relate to supervised algorithms. Existing categorizations of semi-supervised learning methods generally use a subset of these properties and are typically relatively flat, thereby failing to capture similarities between different groups of methods. Furthermore, the categorizations are often fine-tuned towards existing work, making them less suited for the inclusion of new approaches. Figure 1 gives a general taxonomy of the semi-supervised learning methods.

In this project, we use the generative method in the semi-supervised learning. To be more specific, we use GAN-BERT model, which combines the BERT training with GAN model in a scene that we train the unlabeled data in a generative adversarial setting. That is, a generator produces "fake" examples resembling the data distribution, while BERT is used as a discriminator.

The organization of the rest of the paper is as follows. Section 2 gives a literature review of some other related papers working on NLP in semi-supervised learning. Section 3 provides an introduction to GAN. Section 4 talks about GAN-BERT model and how it works. Section 5 gives two experiments by using IMDB datasets and SNLI datasets and analyzes the result. Section 6 gives the conclusion and offers closing remarks.

2 Literature Review

Numerous deep learning models have been proposed in the past few decades for text classification. Based on the disserent input datasets analyzed to classify the data, these methods consist of a single-label, multi-label, unsupervised, semi-supervised

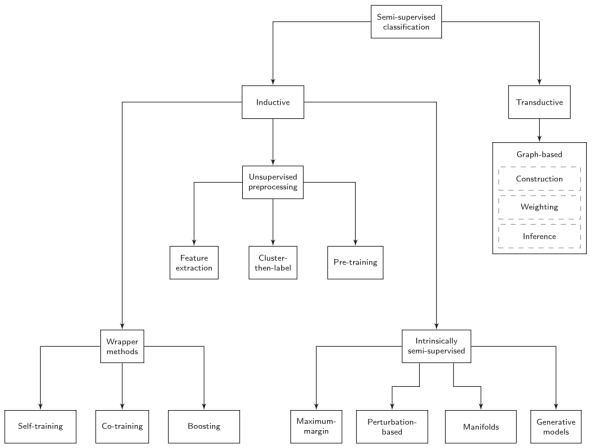

Figure 1: A general taxonomy of the semi-supervised learning methods

and so on. Here we mainly focus on NLP model on the semi-supervised learning. We need to mention that there are a lot of methods and algorithms appearing in the recent years, here we just select some of them which are similar to our idea and could be representative for BERT-based and GAN-based models.

From the perspective of BERT, one useful method is called DistilBERT (Sanh et al., 2019), which is a method to pretrain a smaller general-purpose language representation model and then can be fine-tuned with good performances on a wide range of tasks like its larger counterparts. Based on the technique of knowledge distillation, in which a compact model is trained to reproduce the behaviour of a larger model or an ensemble of models, the token-type embeddings and the pooler are removed while the number of layers is reduced by a factor of 2. It showed that DistilBERT is 40% smaller, 60% faster, and retains 97% of the language understanding capabilities compared to BERT.

SpanBERT (Joshi et al., 2020) is specially designed to better represent and predict spans of text, as shown in Figure 2. It optimizes BERT from three aspects and achieves good results in multiple tasks such as Question Answering. The specific optimization is embodied in three aspects. Firstly, the span mask scheme is proposed to mask a continuous paragraph of text randomly. Secondly, Span Boundary Objective (SBO) is added to predict span by the token next to the span boundary to get the better performance to fine-tune stage. Thirdly, the NSP pre-training task is removed.

For GANs, one effective semi-supervised

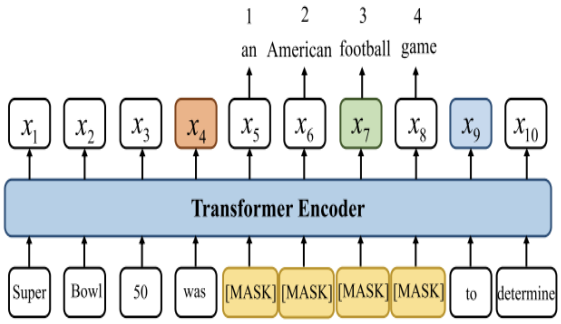

Figure 2: The architecture of SpanBERT

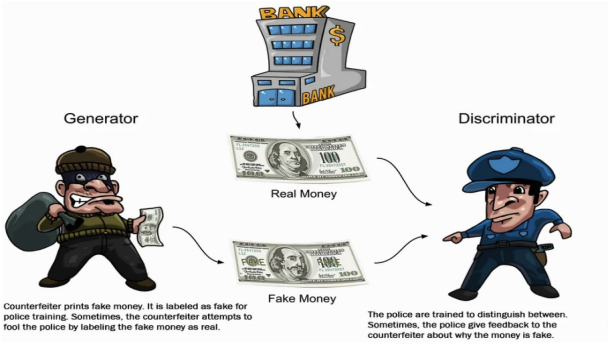

Figure 3: A scenario of the counterfeiters and the police

method is Semi-Supervised Generative Adversarial Networks (SS-GANs)(Salimans et al., 2016). SS-GANs are an extension to GANs where the discriminator also assigns a category to each example while discriminating whether it was automatically generated or not. In SS-GANs, the labeled material is thus used to train the discriminator, while the unlabeled examples (as well as the ones automatically generated) improve its inner representations.

For GAN-BERT, there are some papers working on its application on some scenario. For example, (Ta et al., 2022) address the task of paraphrase identification in Mexican Spanish (PAR-MEX) at a sentence level. They used GAN-BERT just to see how well it can be for their targeted problem and it shows that GAN-BERT reached $F1$ accuracy as 91%.

3 Generative Adversarial Nets

In this section, we would briefly talk about the basic Generative Adversarial Nets (GAN)(Goodfellow et al., 2014) model, which was introduced by. GAN are often used for unsupervised learning tasks, eliminating the need for labeled real data during the training process, which is helpful for us to think about semi-supervised learning problems.

Imagine a scenario where there are two groups of people: the counterfeiters and the police. The counterfeiters are producing fake currency, and the police is trying to detect the counterfeit's fake currency. For both of them, their ability of producing and distinguishing would be strengthened with the time goes. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable. Figure 3 below describes this scene.

This is a vivid example of the Generative Adversarial Nets (GAN). Actually, in the proposed adversarial nets framework, there is a competition between the generative model and its adversary. There is a discriminative model D that learns to determine whether a sample is from the model distribution or the data distribution, and another model, which is generative model G, can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency.

To be more formulaic, these two models D and G play the following two-player minimax game with value function $V(G, D)$:

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

For the formula above, we define a prior distribution on the input noise variable $p_z(z)$ to learn the generator's distribution over data x, then represent a mapping to data space as $G(z;\theta_g)$, where G is a differentiable function represented by a multilayer perceptron with parameters θ_g. For the discriminative model D, we also define a second multilayer perceptron $D(x;\theta_d)$ that outputs a single scalar. Here, $D(x)$ represents the probability that x came from the data rather than the noise p_g. To find the optimal model D and G, we use a minimax optimization scheme, which means we train the discriminative model D to maximize the probability of assigning the correct label to both training examples and samples generated from G by employing the cross-entropy loss. Simultaneously, we train G to interrupt the growth of the discriminative ability of D by minimizing the last term $log(1 - D(G(z)))$.

Figure 4 gives an example for the transition of the model D and G in GAN. Here the blue dashed line represents the discriminative distribution D, the black solid line represents the data distribution

p_x. and the green solid line represents the generative distribution $p_g(G)$. The lower horizontal line is the domain from which the noise z is sampled, and here in this case it is uniformly distributed. The horizontal line above is part of the domain of x. The upward arrows show how the mapping $x = G(z)$ imposes the non-uniform distribution p_g on transformed samples. G contracts in regions of high density and expands in regions of low density of p_g. From the figure we can see the transition of these two models. At the beginning, D can distinguish two distributions partially accurately, giving a high probability for the data distribution and a low probability for the generated distribution and there is some uncertainty in the right tail. Then we train the model D to discriminate samples from data as in (b). (c) shows the update of the model G make G gets closer to the region that are more likely to be classified as real data. And finally, after several steps of training, the model G and D would come to a point at which both cannot improve. In this scene, now we have $p_g = p_{data}$. What's more, the discriminator is unable to differentiate between these two distributions, giving a probability of $D(x) = 1/2$ for all points in this plot.

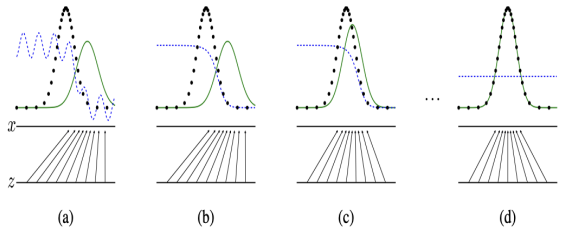

Figure 4: An transition of D and G in GAN training

In practice, GAN algorithm implements the game using an iterative, numerical approach. We first optimize the discriminative model D by k steps and then alternatively optimize the model G by one step. We use this way because if we optimize D to completion and then train G, it firstly will bring a large number of computation, and on the finite data it would result in overfit for our data. By using an alternative thought, it will maintain D to be near its optimal solution, as long as G changes slowly enough. The algorithm procedure is formally presented in Figure 5.

One more thing need to notice is that instead of training G to minimize $log(1 - D(G(z)))$, we can train G to maximize $logD(G(z))$. This objective function results in the same fixed point of the dynamics of G and D but provides much stronger gradients early in learning.

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Figure 5: Minibatch stochastic gradient descent algorithm of GAN

In general, GAN employs a generator and a discriminator in an adversarial training setup, where they compete against each other. The generator aims to produce realistic data, while the discriminator attempts to distinguish between real and generated data. Some advantages of GAN are that GAN can produce highly realistic data, finding applications in artistic creation, image synthesis, and beyond. Also, GAN excels in unsupervised learning tasks, providing valuable data generation without the need for extensive labeled real data.

4 GAN-BERT

In this section, we would give a detailed description of the model we used, which is called GAN-BERT model (Croce et al., 2020). It extends the BERT training with unlabeled data in a generative adversarial setting for semi-supervised learning problems.

Before talking about GAN-BERT, we first give a brief introduction to the semi-supervised GANs (SS-GANs).

SS-GANs enable the semi-supervised learning in a GAN framework. Specifically, a discriminator D is trained over a $(k + 1)$-class objective: "true" examples are classified in one of the target $(1, ..., k)$ classes, while the generated samples are classified into the $k + 1$th class. With the aim of training a semi-supervised k-class classifier, there are some improvements on the discriminative model. Let $p_{fake} = p_m(\hat{y} = y|x, y = k+1)$ be the probability provided by the model m that a generic example x is in the $(k + 1)$th class and $p_{real} = p_m(\hat{y} = y|x, y \in \{1, ..., k\})$ be the probability that x is

real, belonging to one of the real classes. Then the loss function of D is defined as: $L(D) = L_{D_{sup}} + L_{D_{unsup}}$, where:

$$L_{D_{sup}} = -\mathbb{E}_{x,y\sim p_{data}} log[p_{real}]$$
$$L_{D_{unsup}} = -\mathbb{E}_{x\sim p_{data}} log[1 - p_{fake}] - \mathbb{E}_{x\sim p_z} log[p_{fake}].$$

Here, $L_{D_{sup}}$ measures the error in assigning the wrong class to a real example among the original k categories. $L_{D_{unsup}}$ measures the error in incorrectly recognizing a real (unlabeled) example as fake and not recognizing a fake example.

Simultaneously, G is trying to generate examples that are similar to the ones sampled from the real distribution, which means that the average example generated in a batch by G should be similar to the real one. Formally, let's $f(x)$ denote the activation on an intermediate layer of D. The feature matching loss of G is then defined as:

$$L_{G_{fm}} = ||\mathbb{E}_{x\sim p_{data}} f(x) - \mathbb{E}_{x\sim p_z} f(x)||_2^2.$$

Also, the loss of the generative model should consist of the error induced by fake examples identified by D:

$$L_{G_{unsup}} = -\mathbb{E}_{x\sim p_z} log[1 - p_{fake}].$$

And the loss of G is $L(G) = L_{G_{fm}} + L_{G_{unsup}}$.

As the name of GAN-BERT, this model combines BERT and SS-GANs for the fine-tuning stage. By using a pre-trained BERT model, we adapt the fine-tune it by adding the SS-GANs layers to enable semi-supervised learning. To be more specific, given an input sentence $s = (t_1, ..., t_n)$, BERT produces output $n + 2$ vector representations, i.e. $(h_{CLS}, h_{t_1}, ..., h_{t_n}, h_{SEP})$, where h_{CLS} and h_{SEP}) representation are the begin and end of a sentence.

As shown in Figure 9, there is a discriminative model D to classify the examples and a generative model G to act adversarially, which output a vector $h_{fake} \in \mathcal{R}^d$. The discriminator receives the vector, which might be fake or unlabeled or labeled examples from the real distribution. And we want D can classify them in one of the k categories if the vector is from real examples, and classify them as $k + 1$th category if they are fake. This can be achieved by optimizing the two competing losses L_D and L_G in the algorithm scheme of GAN.

After training, G would be discarded while retaining the rest of the original BERT model for inference, which means that there is no additional cost at inference time with respect to the standard BERT model. In this way, GAN-BERT model exploits both the capability of BERT to produce high-quality representations of input texts and to adopt unlabeled material to help the network in generalizing its representations for the final tasks.

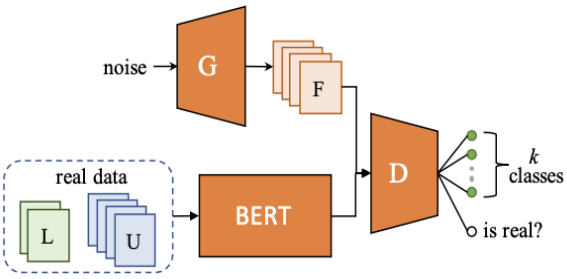

Figure 6: GAN-BERT architecture

5 Experiment

In our experiment, we utilized the GAN-BERT model for a comprehensive analysis and evaluation of two distinct datasets: IMDB and SNLI. The analysis was conducted under the premise of limited labeled data, where only 1% of the dataset was labeled, and the remaining 99% of the data was unlabeled. During training, the unlabeled data was treated by assigning them a pseudo-label of "Unknown,", as shown in the figure below, allowing the model to differentiate between labeled and unlabeled instances. This approach aimed to explore the GAN-BERT model's efficacy and robustness in handling scenarios with minimal labeled data while leveraging the power of semi-supervised learning techniques.

	review	sentiment
37692	First of all, the genre of this movie isn't co...	UNK
26859	This film recreates Lindbergh's historic fligh...	UNK
28567	Although Casper van Dien and Michael Rooker ar...	UNK
25079	I liked this movie. I wasn't really sure what ...	UNK
18707	Yes non-Singaporean's can't see what's the big...	UNK

Figure 7: 99% Unlabeled Data

5.1 Experiment 1: GAN-BERT Training on IMDB

Approach: The GAN-BERT model was trained on the IMDB dataset for sentiment analysis. The training approach involved using 1% of the labeled data for GAN-BERT training shown as the figure below, with the remaining data left unlabeled. The

Transformer used in this experiment was 'bert-base-cased'. The Generator and Discriminator were designed and trained to leverage the GAN framework.

```
df_train_for_ganbert = df_train.sample(frac=0.01) # use 1% of the labeled data for training

df_unlabeled= df_train.drop(df_train_for_ganbert.index)

print(df_train_for_ganbert.shape, df_unlabeled.shape)

(397, 2) (39326, 2)
```

Figure 8: 1% Labeled Data

Hyperparameters of the Best Model:

- Transformer: 'bert-base-cased'
- Number of hidden layers in Generator: 1
- Number of hidden layers in Discriminator: 1
- Noise size: 100
- Dropout rate for Generator and Discriminator: 0.2
- Learning rate for Discriminator: 5×10^{-5}
- Learning rate for Generator: 5×10^{-5}
- Number of training epochs: 3
- Batch size: 32

Accuracy on Training/Test Sets:

- **Training Set Accuracy**: 74.9%
- **Test Set Accuracy**: 74.6%

5.2 Experiment 2: GAN-BERT Training on SNLI Dataset

Approach: The GAN-BERT approach utilizes Generative Adversarial Networks (GANs) with BERT for training on the SNLI (Stanford Natural Language Inference) dataset. The model architecture comprises a generator and a discriminator, where the generator aims to generate fake data with a distribution similar to the encoded real data from BERT. Meanwhile, the discriminator is responsible for distinguishing between real and fake data generated by the generator.

Hyperparameters of the Best Model: The hyperparameters used for the best-performing GAN-BERT model on the SNLI dataset are as follows:

- Maximum Sequence Length: 128
- Batch Size: 8
- Number of Hidden Layers in Generator: 1
- Number of Hidden Layers in Discriminator: 1
- Noise Size: 100
- Dropout Rate for Generator: 0.2
- Learning Rate for Discriminator: 5×10^{-5}
- Learning Rate for Generator: 5×10^{-5}
- Number of Training Epochs: 1
- Multi-GPU: Enabled

Performance Metrics: The GAN-BERT model achieved the following accuracy on different datasets:

- Training Set Accuracy: 32.9%
- Test Set Accuracy: 32.7%

5.3 Result

The experiment result can be seen below:

Accuracy	BERT	GAN-BERT (1% label data)
IMDB Dataset	81.7%	74.6%
SNLI Dataset	84.3%	32.9%

Figure 9: Experiment Result

5.4 Analysis

The GAN-BERT models, trained on both the IMDB and SNLI datasets using only 1% labeled data, displayed distinct performances.

For the IMDB dataset, the GAN-BERT model showcased a remarkable improvement in accuracy, achieving approximately 74.9% and 74.6% accuracy on the training and test sets, respectively. Despite the initial challenges posed by limited labeled data, employing the same training parameters as the pre-trained BERT model led to a significant leap in performance. Leveraging a semi-supervised learning approach, coupled with the power of the BERT-based Transformer and GAN architecture, resulted in more robust representations and enhanced accuracy in sentiment analysis tasks.

However, challenges persist, primarily stemming from the scarcity of labeled data, which might hinder the model from achieving even higher accuracy levels. Moreover, the inherent complexities associated with GANs and the limited number of training

epochs could limit the model's convergence and its ability to comprehend intricate sentiment patterns comprehensively.

Turning to the SNLI dataset, the GAN-BERT model's performance was comparatively lower, achieving around 32.9% and 32.7% accuracy on the training and test sets, respectively. The model's limited performance can be attributed to the significantly reduced labeled dataset (1%) utilized for training, restricting its learning capacity to capture complex patterns inherent in the SNLI dataset. Training for a single epoch may have further limited the model's convergence and learning.

The lower accuracy observed in the SNLI dataset aligns with expectations due to the scarcity of labeled data. The challenges posed by limited labeled data make it difficult for GAN-BERT to match the performance of a pre-trained BERT model trained on a more extensive labeled dataset.

Future experiments could explore increasing the labeled data proportion, extending the training duration, and fine-tuning hyperparameters for both datasets. These endeavors aim to empower GAN-BERT to capture richer representations and enhance predictive performance on sentiment analysis and natural language inference tasks. Effectively leveraging GANs for semi-supervised learning with limited labeled data remains a focus area for improving GAN-BERT's performance across different tasks.

6 Conclusions and Future Work

This study delved into the application of the GAN-BERT model to explore the realm of natural language processing (NLP) within semi-supervised learning contexts. By combining the strengths of Generative Adversarial Networks (GANs) and BERT (Bidirectional Encoder Representations from Transformers), both renowned for their successes in diverse NLP tasks, this research aimed to investigate their collective potential in scenarios characterized by limited labeled data.

Our experiments, conducted on both the IMDB sentiment analysis dataset and the Stanford Natural Language Inference (SNLI) dataset, revealed notable insights into the performance of the GAN-BERT model in semi-supervised settings.

For the IMDB dataset, leveraging only 1% of labeled data, the GAN-BERT model showcased impressive results, achieving approximately 74.9% and 74.6% accuracy on the training and test sets, respectively. Despite the constraints of minimal labeled data, aligning with the training parameters of the pre-trained BERT model notably enhanced the model's performance. The utilization of a semi-supervised learning approach, along with the power of BERT-based Transformers and GAN architecture, facilitated the creation of robust representations, significantly enhancing accuracy in sentiment analysis tasks.

However, challenges persist in both datasets, predominantly stemming from the scarcity of labeled data, potentially limiting the models from achieving higher accuracy levels. Additionally, the complexities inherent in GANs, coupled with a constrained number of training epochs, might impede the model's convergence and comprehensive understanding of intricate sentiment patterns.

In contrast, the SNLI dataset exhibited comparatively lower accuracy with the GAN-BERT model, achieving around 32.9% and 32.7% accuracy on the training and test sets, respectively. The model's limitations can be traced back to the significantly reduced labeled dataset (1%), restricting its ability to grasp the intricate patterns inherent in the SNLI dataset. Furthermore, training for a single epoch may have constrained the model's convergence and learning potential.

In order to enhance the performance on the SNLI dataset, additional efforts can be dedicated to further fine-tuning the GAN-BERT model. This could involve a series of potential strategies:

1. **Extended Training Duration:** Increasing the training time by conducting more epochs or longer training sessions could allow the model to converge better and capture more intricate patterns present in the SNLI dataset.

2. **Hyperparameter Tuning:** Systematically exploring and optimizing the hyperparameters of the GAN-BERT model, such as learning rates, batch sizes, dropout rates, or other architectural configurations, could significantly impact the model's performance on the SNLI dataset.

3. **Architecture Modifications:** Evaluating and implementing architectural modifications tailored specifically to address the nature of the SNLI dataset might yield improvements. This could involve alterations to the model's layers and attention mechanisms, or incorporating additional contextual information.

4. **Enhanced Tokenization Techniques:** Improving the tokenization process by utilizing more sophisticated tokenizers or preprocessing techniques specifically tailored for the linguistic complexity found in the SNLI dataset could potentially refine the model's understanding of the text.

5. **Ensemble Methods:** Exploring ensemble techniques by combining multiple GAN-BERT models or incorporating other complementary models could leverage diverse perspectives and improve the overall performance of the SNLI dataset.

By iteratively refining the GAN-BERT model through these avenues, we aim to boost its ability to capture nuanced semantic relationships and subtleties inherent in natural language inference tasks, thereby achieving higher accuracy and robustness on the SNLI dataset.

References

Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 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*, pages 2114–2119, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2016. A neural network for coordination boundary prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 23–32, Austin, Texas. Association for Computational Linguistics.

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.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Adaptive semi-supervised learning for cross-domain sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3467–3476, Brussels, Belgium. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(2):1–41.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Hoang Thang Ta, Abu Bakar Siddiqur Rahman, Lotfollah Najjar, and Alexander Gelbukh. 2022. Gan-bert, an adversarial learning architecture for paraphrase identification. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022)*.

Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440.