Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

# GAN-BElectra: Enhanced Multi-class Sentiment Analysis with Limited Labeled Data

**Md. Riyadh & M. Omair Shafiq**

**To cite this article:** Md. Riyadh & M. Omair Shafiq (2022) GAN-BElectra: Enhanced Multi-class Sentiment Analysis with Limited Labeled Data, Applied Artificial Intelligence, 36:1, 2083794, DOI: 10.1080/08839514.2022.2083794

**To link to this article:** https://doi.org/10.1080/08839514.2022.2083794



Published online: 26 Jun 2022.

APPLIED ARTIFICIAL INTELLIGENCE
2022, VOL. 36, NO. 1, e2083794 (2315 pages)
https://doi.org/10.1080/08839514.2022.2083794

Taylor & Francis
Taylor & Francis Group



# GAN-BElectra: Enhanced Multi-class Sentiment Analysis with Limited Labeled Data

Md. Riyadh and M. Omair Shafiq

School of Information Technology, Carleton University, Ottawa, Ontario, Canada

**ABSTRACT**

Performing sentiment analysis with high accuracy using machine-learning techniques requires a large quantity of training data. However, getting access to such a large quantity of labeled data for specific domains can be expensive and time-consuming. These warrant developing more efficient techniques that can perform sentiment analysis with high accuracy with a few labeled training data. In this paper, we aim to address this problem with our proposed novel sentiment analysis technique, named GAN-BElectra. With rigorous experiments, we demonstrate that GAN-BElectra outperforms its baseline technique in terms of multiclass sentiment analysis accuracy with a few labeled data while maintaining an architecture with reduced complexity compared to its predecessor.

**ARTICLE HISTORY**
Received 5 February 2022
Revised 23 May 2022
Accepted 25 May 2022

## Introduction

In recent times, high-performance computing along with cloud technologies have helped many research domains flourish rapidly. One such domain is machine learning and applied research based on machine-learning techniques such as self-driving cars and language translation services (Zhang, Wang, and Liu 2018). The dramatic rise in Internet usage and social media applications have made most people with a connected device a content generator of some sort. These user-generated contents are essential in the growth of many machine-learning-based applications such as virtual personal assistants and recommender systems (Ricci, Shapira, and Rokach 2015). The influx of user generated content elevated the demand for efficient techniques to analyze them to generate valuable insights for government and companies alike. This also caught the attention of researchers who are designing increasingly efficient and accurate data analysis techniques in a frequent manner. Natural Language Processing (NLP) of textual data is one such data analysis domain that saw rapid innovation in recent times. NLP's general goal is to create an understanding of language in computers (Liddy 2001). Identifying sentiment

**CONTACT** Md. Riyadh mdriyadh@cmail.carleton.ca; M. Omair Shafiq omair.shafiq@carleton.ca School of Information Technology, Carleton University, Ottawa, Ontario, Canada

expressed in a text is a task that contributes to that overall understanding. This task is generally known as Sentiment Analysis, a research area that has seen rapid progress in recent years (Patel 2015; Zhang, Wang, and Liu 2018).

With sentiment analysis, a given piece of text is categorized with respect to the sentiment it expresses, often in terms of positive, neutral, or negative. It commonly uses machine-learning techniques in order to identify such patterns and then perform these categorizations of textual data. This type of machine learning technique, commonly referred to as supervised learning, heavily relies on training data. Sentiment analysis is a domain-specific task. This means words used in one context to express certain sentiments may exert different sentiments in different contexts. As a result, training a machine-learning model for sentiment analysis tasks typically requires training with domain-specific labeled data. Having a large quantity of labeled training data for specific domains can be tedious and expensive. In this research, we focus on the problem of the limited labeled training data in sentiment analysis.

Machine-learning-based sentiment analysis models trained with a low number of training data can perform poorly on this text classification task. However, since a large number of training data may not be always available, researchers have attempted to develop techniques to have higher sentiment classification accuracy with a lower number of labeled training data. These techniques, however, still fall short in reaching the peak performance of machine learning models trained on large labeled datasets. As a result, achieving higher accuracy in sentiment classification task with low amount of training data is still a worthwhile research problem to address.

In this study, we propose a novel sentiment classification technique named GAN-BElectra that aims to attain high accuracy in sentiment classification with a few labeled data (i.e., 50 labeled training datapoints per class). GAN-BElectra focuses on multi-class sentiment analysis. In this type of sentiment analysis, there are more than two sentiment classes to choose from. The typical class labels are positive, neutral, and negative. This is in contrast with binary classification where only two sentiment classes are used for categorization: positive and negative. Multi-class sentiment analysis, especially the ones that consider neutral as one of the classes, seems to possess more practical value compared to binary classification since not every text expresses a sentiment, and some texts naturally fall into the neutral category.

GAN-BElectra outperforms our recently published technique named SG-Elect (Riyadh and Shafiq 2021). SG-Elect attempted to address the same that is the main focus of this paper – multiclass sentiment analysis with limited labeled data. SG-Elect employs three machine-learning components. Two of them are deep-learning components (GAN-BERT (Croce, Castellucci, and Basili 2020) and Electra (Clark et al. 2020)) and the other one is a traditional machine-learning component (Semi-Supervised Self Trainer (Yarowsky 1995). In GAN-BElectra, which is an extension of SG-Elect, we employ only two

deep-learning components (GAN-BERT and Electra), which significantly reduces the complexity of the architecture while still achieving higher sentiment classification accuracy than SG-Elect.

As part of this research, we make the following contributions:

- We propose a novel technique for multi-class sentiment analysis named GAN-BElectra which builds upon SG-Elect and achieves higher classification accuracy than its predecessor (i.e., SG-Elect) with a few labeled training data while maintaining a more lightweight architecture.
- We evaluate GAN-BElectra on three datasets that are available in the public domain. These experiments illustrate that GAN-BElectra outperforms its state-of-the-art (SOTA) baseline (i.e., SG-Elect) in the multi-class sentiment analysis task with limited labeled training data.
- We dissect GAN-BElectra's architecture and analyze the individual components that constitute it to facilitate a thorough understanding of the proposed solution.

## Related Work

The lack of labeled data is a common problem in text classification tasks. There are several proposals in the literature to address this general problem (Croce, Castellucci, and Basili 2020; Liu et al. 2015; Miao et al. 2020). Using lexicons to classify texts is a typical solution offered for this issue. (Mazharul Islam, Dong, and de Melo 2020). An example of this is technique shown in Figure 1. However, the lack of accuracy of lexicon-based techniques in text classification, especially in comparison to fully supervised machine-learning-based methods is well known (Hemmatian and Karim Sohrabi 2019).

Semi-supervised learning is another common technique that attempts to resolve the issue of lack of labeled training data. This method is specially engineered to train machine-learning models with a few labeled data (Kumar, Packer, and Koller 2010; Mastoropoulou 2019). Researchers have designed several variations of semi-supervised learning. These include using teacher-student method where teacher confidence is utilized to identify easy samples during training (e.g., self-paced co-training (Fan et al. 2017), self-paced learning (Kumar, Packer, and Koller 2010)). Other examples include utilizing meta-


**Figure 1.** Example of basic lexicon-based sentiment analysis.

learning (Xinzhe et al. 2019) and active learning (Mastoropoulou 2019) for sample selection based on teacher confidence. A recent application of semi-supervised learning, specifically for sentiment analysis, is a neural network-based semi-supervised learning framework proposed by Li et al. (Ning, Chow, and Zhang 2020). It performs training with a few labeled data along with many unlabeled data. To address the labeled data scarcity issue in short text classification tasks such as sentiment analysis of tweets, Yang et al. (Yang et al. 2021) proposed a heterogeneous graph attention network that embedded a flexible heterogeneous information network framework that modeled short text with the functionality to include additional information while appropriately detecting their semantic relations. By extending their technique with semi-supervised inductive learning, they demonstrate that the proposed solution outperforms their selected SOTA baselines for both single and multi-label classification tasks. Kim et al. (Kim, Son, and Han 2022) proposed a novel self-training method that leveraged a lexicon to guide its mechanism in generating pseudo-labels in order to address the lack of labeled data in text classification, particularly sentiment analysis. They demonstrated that the guidance from lexicon in their experimental setup enhanced the reliability of pseudo labels by performing manipulation on the loss term.

Text augmentation is another method to mitigate the labeled data scarcity issue in text classification. Abonizio et al. (Abonizio, Cabrera Paraiso, and Barbon Junior 2021) conducted an in-depth study of the usage of text augmentation in addressing labeled data scarcity issues in sentiment analysis. They offered a taxonomy for these techniques, first categorizing all techniques into "sentence" manipulation and "embedding" manipulation (i.e., manipulating representative vectors of text instead of the actual text data), and then further dividing the "sentence" manipulation category into three subcategories: transformation (e.g., synonym replacement), paraphrasing (e.g., adapting translation models to generate rephrased text in the same language), and generation (e.g., using autoregressive language models such as GPT2 (Radford et al. 2019) to generate text). They evaluated various text augmentation techniques and observed how they influenced the sentiment classification accuracy of different techniques in scenarios such as a low number of labeled training data. For instance, they found that BERT (Devlin et al. 2018) and ERNIE (Sun et al. 2019) achieved superior classification performance with a low number of available training samples when boosted with back-translation (Edunov et al. 2018) augmentation technique. Some researchers also applied text augmentation technique to mitigate the labeled data scarcity in non-English languages. For example, Barriere et al. (Barriere and Balahur 2020) used automatic translation of English tweets to French, Spanish, German, and Italian to apply data-augmentation which improved the sentiment analysis performance over non-English tweets using different transformer-based

techniques (Wolf et al. 2019). Edwards et al. (Edwards et al. 2021) demonstrated how leveraging GPT-2 (Radford et al. 2019) driven text augmentation in few-shot learning setup could enhance the text classification accuracy.

Recent advances in pre-trained model made their landfall in the sentiment analysis research area inevitable. Examples of such pre-trained models include BERT (Devlin et al. 2018), Electra (Clark et al. 2020) etc. These models, which are typically based on Transformers (Ashish et al. 2017), are trained on vast amount of data following a self-supervised approach. This provides these models with a general capability to comprehend language. At this stage, these pre-trained models are capable of many tasks that requires understanding of general-purpose language representation. For leveraging them in downstream tasks that require domain specific knowledge, for example sentiment analysis, they go through another light-weight training process known as fine-tuning. This is essentially training the pretrained model with the task/domain specific labeled data. Since the pretrained model already has an understanding of language in general, fine-tuning on a very small set of domain-specific labeled data can make them significantly more accurate as classifiers compared to typical training of machine learning models. As a result, there has been some interest in the researcher community in using these pretrained models to address the issue labeled data scarcity in text-classification task in general. For example, Croce et al. (Croce, Castellucci, and Basili 2020) experimented using semi-supervised generative adversarial network (GAN) to improve BERT's fine-tuning stage. This architecture, named GAN-BERT (Croce, Castellucci, and Basili 2020), was mainly focused on achieving higher accuracy compared to original BERT network in text classification tasks with low amount of labeled training data.

Building upon GAN-BERT, Riyadh et al. proposed SG-Elect (Riyadh and Shafiq 2021), which utilizes GAN-BERT in its architecture for pseudolabel generation along with another traditional machine-learning-based self-training mechanism ("Self-Training – Learn Documentation" n.d.). Their architecture also includes an Electra-based (Clark et al. 2020) final classifier. Together with these components, SG-Elect achieves higher accuracy in multi-class sentiment analysis task compared to GAN-BERT for three publicly available datasets.

Researchers have proposed various approaches that attained decent accuracy in sentiment classification task with limited labeled data. While approaches that rely on a lot of training data can achieve considerably higher accuracy in this task, achieving similar accuracy with limited labeled data still remains a challenge. As a result, achieving higher accuracy in sentiment classification task using a few labeled data is still a worthwhile investigation.

## Proposed Solution

Our proposed solution, GAN-BElectra consists of two primary components: GAN-BERT (Croce, Castellucci, and Basili 2020) and Electra (Clark et al. 2020). GAN-BERT acts as a pseudolabel generator in GAN-BElectra, whereas Electra consumes those pseudolabels to fine-tune itself. GAN-BElectra builds upon our previously designed technique, SG-Elect (Riyadh and Shafiq 2021) which attempted to address the same problem of multiclass sentiment analysis with limited labeled data. SG-Elect's architecture has more complexity compared to GAN-BElectra as it contains an additional machine-learning component – semi-supervised self-trainer. Below we discuss the components of GAN-BElectra in greater details.

### *GAN-BERT*

GAN-BERT is a deep-learning network which contains BERT (Bidirectional Encoder Representations from Transformers) (Fan et al. 2017), and SS-GAN (Semi-supervised Generative Adversarial Network) (Croce, Castellucci, and Basili 2020). BERT is a pretrained model based on Transformer technology (Ashish et al. 2017). It learns contextual relations between words using an attention-based mechanism. The model was pretrained with large training data consisting of raw texts. The pre-trained model is then fine-tuned on training data for specific task.

GAN-BERT's novelty mainly stem from the fact that it extends BERT's fine-tuning phase with an SS-GAN. SS-GAN consists of two main parts: a) a Generator and b) a Discriminator as shown in Figure 2. The Generator generates synthetic labels, and the Discriminator classifies those labels into real and fake using adversarial learning. In GAN-BERT, a pre-trained BERT model is fine-tuned with task-specific layers first, which is part of BERT's typical fine-tuning process. Second, during this fine-tuning stage, SS-GAN layers is used to enable semi-supervised learning.

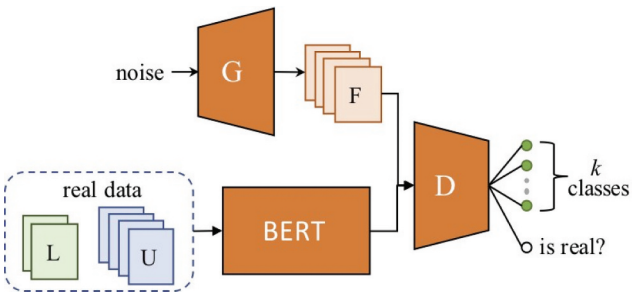


**Figure 2.** GAN-BERT Architecture (Croce, Castellucci, and Basili 2020). U, L, F, G, D denote unlabeled data, labeled data, fake labels, generator, and discriminator respectively.

We use GAN-BERT in its original configuration from the study (Croce, Castellucci, and Basili 2020). At first, we use GAN-BERT's fine-tuning process to fine-tune the model with the original few labeled data. Next, the fine-tuned GAN-BERT model is used to generate pseudolabels for the unlabeled data.

The next component in our architecture is Electra which takes the pseudo-label output from GAN-BERT as its input. We discuss this component below.

### *Electra*

The final component in our architecture is a transformer-based pretrained model named Electra (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) (Clark et al. 2020) along with typical adjacent deep-learning layers such as input embedding layer, dropout layer, and a dense layer (Figure 3). The Electra pretrained model layer is placed immediately after the input embedding layer. It employs an efficient training technique compared to other MLMs (masked language model). MLMs similar to BERT leverage subsets of unlabeled input for pre-training. In this technique, the transformer network learns to recognize the masked tokens and then retrieves the original input from that token. In contrast, Electra leverages replaced tokens to corrupt the input in rather than using masked subset of input. In the next stage, the deep-learning network is pretrained as a discriminator. The discriminator differentiates between the original token vs. replaced token.

In GAN-BElectra, we take the pooled output from the Electra pretrained model layer to a Dropout layer. This Dropout layer helps the model to avoid overfitting. The subsequent dense layer (deeply connected neural network layer) takes the output from Electra and outputs logits. We then apply Softmax ("Scipy.Special.Softmax – SciPy v1.8.0 Manual" n.d.) and Argmax function ("Numpy.Argmax – NumPy v1.22 Manual" n.d.) to those logits to attain the predicted labels.

This final Electra-based classifier model uses an optimizer named AdamW (Loshchilov and Hutter 2019). It enhances Adam optimizer (Kingma and Jimmy Lei 2015) with improvement to the weight decay implementation through a stochastic optimization mechanism. This includes separating weight decay from the gradient update.

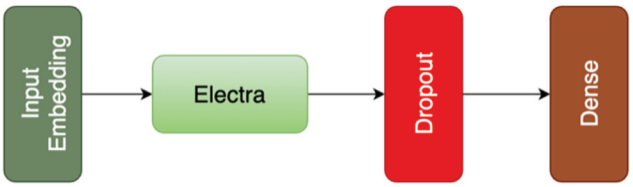


**Figure 3.** Electra-based pretrained component.

The loss calculation for this final classifier is performed using Sparse Categorical Cross-entropy. This selection is based on the fact that we are performing multi-class sentiment analysis which involves categorization of input text into more than two mutually exclusive classes.

The number of epochs (10) is based on the experimental performance of our network across various validation sets. We fine-tine this classifier twice. First with the pseudolabels generated by GAN-BERT. Then, the next fine-tuning is performed using the few original labeled data.

Figure 5 shows the overall architecture of GAN-BElectra. This architecture evolved from SG-Elect (Figure 4) where there was an additional Semi-Supervised Self Training Classifier acted as pseudolabel generator in addition to GAN-BERT.

GAN-BElectra streamlines SG-Elect architecture by stripping away the process-heavy Self Training Classifier, leaving only GAN-BERT as the sole generator of pseudolabels. This also abolishes the need of having the "Combinator" component which further simplifies the architecture.

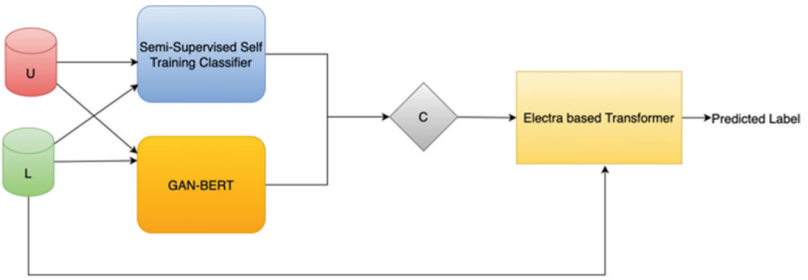


**Figure 4.** SG-Elect architecture (Riyadh and Shafiq 2021), U, L, C denote unlabeled data, labeled data, and combinator respectively.

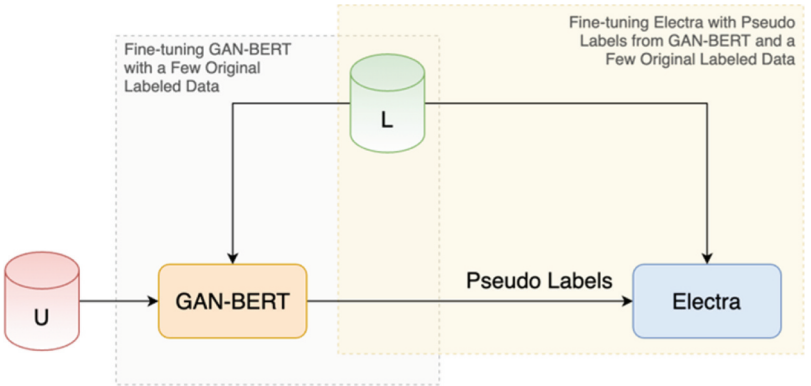


**Figure 5.** GAN-BElectra architecture. U and L denote unlabeled and labeled data respectively.

Algorithm I: Training process of GAN-BElectra based on SG-Elect (Riyadh and Shafiq 2021)

**inputs:** $\mathbf{X = X_L + X_U}$
$\mathbf{X_L = x_{L1}, x_{L2}, \ldots, x_{Ln}}$
$\mathbf{X_U = x_{U1}, x_{U2}, \ldots, x_{Um}}$
**outputs: Trained model**

```
BEGIN
    for all x ∈ X do {
        C_GB ← Train (GAN-BERT, X)
    } end for
    return C_GB
    for all x_U ∈ X_U do {
        Y_GB ← Predict (C_GB, X_U)
    } end for
    return Y_GB
    for all y_GB ∈ Y_GB do {
        C_EL ← Train (ElectraTransformer, Y_GB)
    } end for
    return C_EL
    for all x_L ∈ X_L do {
        C_EL ← Train (ElectraTransformer, X_L)
    } end for
    return C_EL
END
```

Algorithm I demonstrates GAN-BElectra's high-level training process, which builds upon and enhances SG-Elect's training process (Riyadh and Shafiq 2021) with reduced training steps and network complexity. The training process begins with many unlabeled data ($X_U$) and a few labeled ($X_L$). After being trained on the original few labeled data ($X_L$) in a semi-supervised manner, the GAN-BERT component ($C_{GB}$) generates pseudolabels ($Y_{GB}$) for the unlabeled data. We use this pseudolabels to fine-tune our Electra-based pretrained model ($C_{EL}$). This final Electra-based classifier goes through another fine-tuning using our few labeled data. At this point, GAN-BElectra is fully trained, and we evaluate it using our test data.

## Experiments and Evaluation

### *Procedure*

We have performed several experiments in order to evaluate GAN-BElectra along with our SOTA baseline (i.e., SG-Elect) and other two techniques (GAN-BERT, Electra) acted as subcomponents within the proposed solution. Although individual experiments have some specificities, they all share some common steps. These include:

*1) Data splitting*

One of the initial steps in our experiments involve splitting the dataset into training and test sets. For train-test split, we opted for an 80:20 ratio where 80% of the total data accounted for the training dataset and 20% is

allocated for testing. We also split the training dataset into labeled and unlabeled sets. In all our data splitting exercise, we use specific seed numbers to generate controlled randomness. This makes our experiments reproducible.

Once we split our training dataset between labeled and unlabeled sets, we remove the labels from the unlabeled dataset to serve its intended purpose. Labeled dataset contains 50 datapoints for each sentiment class representing a minute fraction of the total datapoints used across our experiments with various datasets.

*2) Data preprocessing*

After train, test, labeled, and unlabeled datasets are created, we perform additional data pre-processing, which includes removal of stop words as necessary, performing vectorization of the datasets so that they are consumable by the machine-learning algorithms. We describe this in greater details in the section that follows.

*3) Defining model*

The next step is to define the machine-learning model. This involves defining the correct parameters and configuration for the model, in addition to defining the layers that eventually construct the deep-learning networks used in this study.

*4) Training and testing*

The after model definition step, we begin training the model with our training data which is different for each experiment. After the training is complete, we test our trained model with test dataset. We report various metrics including F1-score, accuracy based on our evaluations of the models.

To sum up, each experiment mainly consists of dataset splitting, data preprocessing, defining model, training the model, and evaluating the trained model. To make our experimental findings more reliable, we run each of our experiment three times using three different random seeds, and we report the average results from these experiments. It is noteworthy that since GAN-BElectra builds upon SG-Elect (Riyadh and Shafiq 2021) and the evaluation criteria, datasets, and experimental settings are intentionally identical in these two consecutive studies and they share some of the reported results, especially as it relates to the two common individual sub-components, namely GAN-BERT (Croce, Castellucci, and Basili 2020) and Electra (Clark et al. 2020).

## Tools

Our experiments required leveraging several existing tools. The following are some noteworthy examples:

*1) Software*:

(a) Programming language: We used Python in all our experiments. Python provides many useful libraries for NLP tasks out of the box as well as for machine-learning experiments in general.
(b) Libraries: We used many publicly available software libraries in our experiments as required. Some of the noteworthy ones include Numpy ("NumPy" n.d.) and Matplotlib ("Matplotlib: Python Plotting – Matplotlib 3.4.2 Documentation" n.d.). For deep-learning algorithms, we heavily leveraged TensorFlow ("TensorFlow" n.d.).
(c) Machine-learning models: We make use of the GAN-BERT in its original configuration ("GitHub – Crux82/Ganbert: Enhancing the BERT Training with Semi-Supervised Generative Adversarial Networks" n.d.). This serves as the sole pseudolabel generator in our solution. Our deep-learning component involves the large variant of pre-trained Electra model ("TensorFlow Hub – Electra Large" n. d.).

*2) Hardware:*

Instead of using our on-premise hardware resources, we utilized cloud hardware resources provided as part of the Pro-tier subscription of Google Colab ("Colaboratory – Google" n.d.). This platform provides a browser-based user interface to perform coding, which is especially designed for data science tasks. This is supported by backend resources such as Tensor Processing Unit (TPU) ("Cloud Tensor Processing Units (TPUs) | Google Cloud" n.d.) and Graphics Processing Unit (GPU) ("What Is a GPU? Graphics Processing Units Defined" n.d.). The hardware specification provided by Google Colab Pro varied for different experiments and the runtime engines are dynamically allocated. However, the following specification represents the typical maximum hardware resource we received from Google Colab Pro:

- Compute: Intel(R) Xeon(R) CPU @ 2.30 GHz (Max. available cores: 40)
- Memory: 36 GB
- Disk: 226 GB

## Datasets

In order to perform a robust evaluation of our solution, we chose three publicly available datasets: SST5 dataset (Socher et al. 2013), US Airline dataset ("Twitter US Airline Sentiment | Kaggle" n.d.), and Rosenthal, Farra, and Nakov 2017 dataset (2017) (we refer to this as *SemEval* in this paper). We describe these datasets below:

*1) SST5*

SST5 is a 5-class sentiment analysis dataset. SST stands for Stanford Sentiment Treebank (Socher et al. 2013). Datasets with five or more sentiment classes are typically called fine-grained sentiment datasets. SST5 dataset contains short texts, and each short text is labeled with any of the following 5 sentiment classes:

- Very Positive
- Positive
- Neutral
- Negative
- Very Negative

Figure 6 demonstrates the composition of this dataset.

*2) US Airline*

US Airline dataset ("Twitter US Airline Sentiment | Kaggle" n.d.) is a three-class sentiment analysis dataset. Each datapoint consists of short text which is a real Twitter post about US Airline carriers, and a sentiment label which is either Positive, Neutral, or Negative. Figure 7 shows the composition of this dataset.

*3) SemEval*

SemEval (Rosenthal, Farra, and Nakov 2017) is another three-class sentiment analysis dataset we have used in our experiments. This dataset is composed of short text from real Twitter post and each short text is labeled as either Positive, Neutral, or Negative. Figure 8 shows the composition of this dataset.

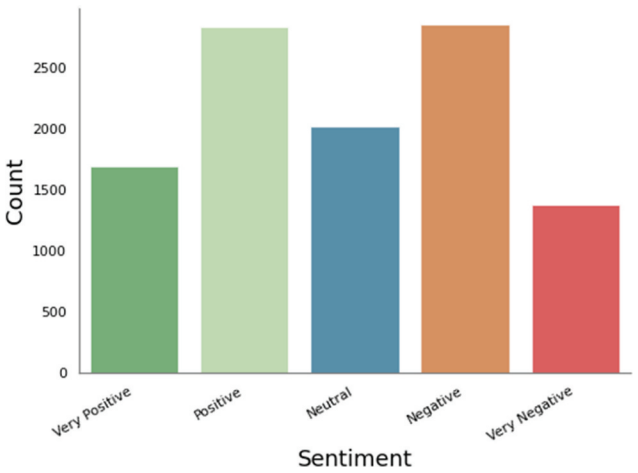


**Figure 6.** SST5 dataset composition.

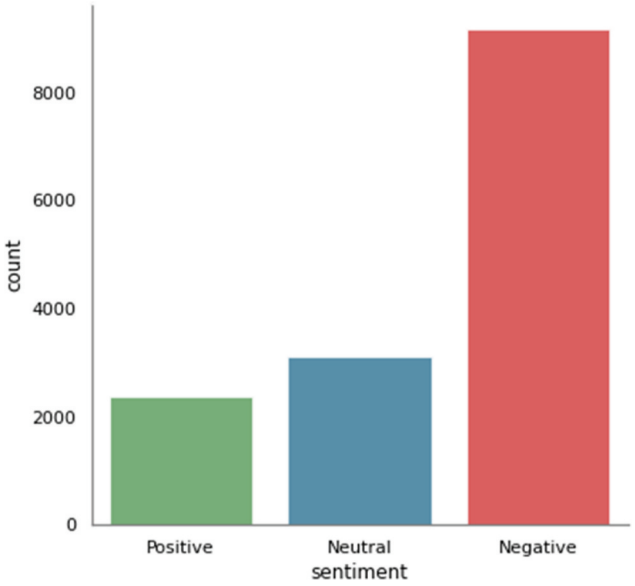


**Figure 7.** US Airline dataset composition.

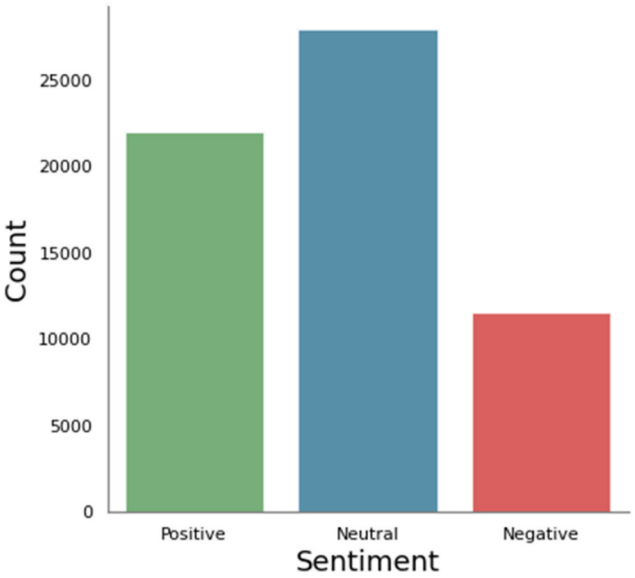


**Figure 8.** SemEval dataset composition.

### *Data Preprocessing*

TensorFlow's BERT tokenizer ("TensorFlow" n.d.) transforms our input texts into numerical format. The deep-learning models incorporated in our experiments anticipate all the inputs sentences concatenated to one another. The

input begins with a "[CLS]" token (this indicates that the task is a "classification problem"). The end of each input is indicated with a separator token "[SEP]."

Finally, the datapoints were represented as Tensors ("TensorFlow" n.d.), and the deep-learning experiments were executed on TPUs ("Cloud Tensor Processing Units (TPUs) | Google Cloud" n.d.)

It is noteworthy that since our usage of 50 datapoints per class as the original few labeled data makes composition of the primary training data organically balanced, we kept the datasets in their original composition without applying any additional balancing technique. Also of note is that the original SemEval dataset contains total 61473 datapoints, majority of which would have been used in our setup as unlabeled data. Since our original labeled data per class is only 50 datapoints, an extensively large amount of unlabeled data provide little value in our setup while significantly increasing the training time. Consequently, we opted to use a representative sample from this dataset (i.e., taking total datapoints 20000 out of 61473, resembling the other three-class dataset used in our study: US Airline dataset) while preserving the original composition (i.e., class distribution) of the dataset by leveraging stratified random sampling (Vries 1986).

### *Evaluated Techniques*

Using the three selected datasets, we evaluated the following techniques:

- GAN-BElectra (Our proposed solution)
- SG-Elect (primary baseline)
- GAN-BERT
- Electra

## Results

As explained in Section IV, in order to make our findings more reliable, we have evaluated each technique (GAN-BElectra, SG-Elect, GAN-BERT, and Electra) three times on each of the three selected datasets using seed numbers to achieve randomized datapoint allocation for train and test split for each evaluation. In summary, we ran a total of 36 experiments which is comprised of nine experiments per technique. We report the average of the results of our evaluations of the techniques on the three selected datasets.

**Table 1.** Summary of Results (F1-Score, Accuracy, Standard Deviation).

| Dataset | Technique | Mean F1-Macro | Mean Accuracy | Standard Deviation of Mean Accuracy |
|---|---|---|---|---|
| SST5 | GAN-BElectra | **0.3796** | **0.3825** | 0.0350 |
| | SG-Elect | 0.3763 | 0.373 | 0.0249 |
| | Electra | 0.3138 | 0.3533 | 0.0225 |
| | GAN-BERT | 0.3185 | 0.3406 | 0.0337 |
| US Airline | GAN-BElectra | **0.6722** | **0.7108** | 0.0195 |
| | SG-Elect | 0.6659 | 0.7072 | 0.0195 |
| | Electra | 0.5493 | 0.623 | 0.0331 |
| | GAN-BERT | 0.6413 | 0.7032 | 0.0514 |
| SemEval | GAN-BElectra | **0.5663** | **0.5741** | 0.0145 |
| | SG-Elect | 0.558 | 0.5635 | 0.0228 |
| | Electra | 0.3992 | 0.4536 | 0.0184 |
| | GAN-BERT | 0.473 | 0.5208 | 0.0448 |

**Table 2.** Summary of Results (Standard Error, Confidence Interval of Standard Error).

| Dataset | Technique | Mean Standard Error | Confidence Interval (at 95%) of Standard Error |
|---|---|---|---|
| SST5 | GAN-BElectra | 0.6175 | 0.0203 |
| | SG-Elect | 0.627 | 0.0202 |
| | Electra | 0.6467 | 0.0199 |
| | GAN-BERT | 0.6594 | 0.0198 |
| US Airline | GAN-BElectra | 0.2892 | 0.0164 |
| | SG-Elect | 0.2928 | 0.0165 |
| | Electra | 0.377 | 0.0176 |
| | GAN-BERT | 0.2968 | 0.0165 |
| SemEval | GAN-BElectra | 0.4259 | 0.0153 |
| | SG-Elect | 0.4365 | 0.0154 |
| | Electra | 0.5464 | 0.0154 |
| | GAN-BERT | 0.4792 | 0.0155 |

Tables 1 and 2 provide the summary of all our experimental results. In Table 1, we report the mean accuracy (i.e., average of three evaluations. Accuracy is calculated using true positives and true negatives as shown in equation I) for each technique on each dataset.

$$accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{I}$$

*where, tp, fp, tn, fn consecutively represent true positive, false positive, true negative, and false negative.*

In addition, we report the standard deviation of this mean, which represents the variations of the accuracy scores across the three evaluations for each technique on each dataset. Table 1 shows that the standard deviation scores for all our experiments were between 0.0195 and 0.0514 indicating high reliability of the reported mean accuracy scores based on our randomized experiments.

Table 1 also contains macro-averaged F1-score (see equation II) which, in contrast with accuracy, provides a more reliable assessment of techniques in the presence of class imbalance (Abonizio, Cabrera Paraiso, and Barbon Junior 2021).

$$F1(macro) = 2 * \frac{pr * re}{pr + re} \quad \text{(II)}$$

*where*

$$pr(precison) = \frac{tp}{tp + fp} \quad \text{(III)}$$

$$re(recall) = \frac{tp}{tp + fn} \quad \text{(IV)}$$

We also report the mean standard error (i.e., inverse of accuracy) in Table 2. Inspired by Tom Mitchell's suggestion for comparing machine-learning models (Mitchell 1997), we additionally report estimation statistics based on the reported standard error at the commonly used significance level of 95%. The following formula was used for the confidence interval (ci) calculation:

$$ci = z \times \sqrt{\frac{e \times (1 - e)}{n}} \quad \text{(V)}$$

*where*

- ***z** is a critical value from the Gaussian distribution which has a value of 1.96 for 95% significance level* (Brownlee 2019),
- ***n** is the size of the test sample,*
- ***e** is the standard error reported in* Table 2.

The equations reported in this chapter (eq. I, II, III, IV, V) can be found in (Brownlee 2019).

Our analysis suggests that the confidence interval for the reported standard error remained between the range of 0.0154 and 0.0203 (see Table 2). This arguably further indicates the robustness of our results across multiple experimental runs for each of the evaluated techniques for the selected datasets.

Table 3 contains our pseudo-label generator's (GAN-BERT) accuracy. Tables 4, 5, and 6 contain detailed results for all evaluated techniques for SST5, US Airline, and SemEval dataset, respectively. Following the tables, we also include confusion matrix visualizing the classification accuracy and true positives and negatives for each evaluated techniques for the three selected datasets. Below we describe our results by dataset.

**Table 3.** GAN-BERT's accuracy in pseudo label generation across three datasets.

| Dataset | GAN-BERT Pseudo Label Accuracy |
|---|---|
| SST5 | 0.3328 |
| US Airline | 0.7087 |
| SemEval | 0.5209 |

**Table 4.** Detailed Results for the SST5 Dataset.

| Technique | Class | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|---|
| GAN-BElectra | Very Negative | 0.3246 | 0.5253 | 0.397 | 0.3825 |
| | Negative | 0.4493 | 0.2461 | 0.3129 | |
| | Neutral | 0.2481 | 0.2601 | 0.2537 | |
| | Positive | 0.4232 | 0.485 | 0.4499 | |
| | Very Positive | 0.4969 | 0.4773 | 0.4847 | |
| SG-Elect | Very Negative | 0.3357 | 0.4854 | 0.3912 | 0.373 |
| | Negative | 0.4604 | 0.2755 | 0.3301 | |
| | Neutral | 0.2617 | 0.3488 | 0.2944 | |
| | Positive | 0.4434 | 0.3452 | 0.3849 | |
| | Very Positive | 0.4695 | 0.5072 | 0.4807 | |
| Electra | Very Negative | 0.4121 | 0.3831 | 0.2701 | 0.3533 |
| | Negative | 0.3828 | 0.2062 | 0.2363 | |
| | Neutral | 0.2385 | 0.3119 | 0.2663 | |
| | Positive | 0.3813 | 0.4459 | 0.4063 | |
| | Very Positive | 0.5194 | 0.4514 | 0.3901 | |
| GAN-BERT | Very Negative | 0.3114 | 0.4908 | 0.3725 | 0.3406 |
| | Negative | 0.4753 | 0.2027 | 0.2476 | |
| | Neutral | 0.2358 | 0.3524 | 0.27 | |
| | Positive | 0.3932 | 0.3564 | 0.3539 | |
| | Very Positive | 0.4881 | 0.4159 | 0.4396 | |

**Table 5.** Detailed Results for the US Airline Dataset.

| Technique | Class | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|---|
| GAN-BElectra | Negative | 0.8896 | 0.7231 | 0.7975 | 0.7108 |
| | Neutral | 0.463 | 0.6516 | 0.5404 | |
| | Positive | 0.6277 | 0.7408 | 0.6788 | |
| SG-Elect | Negative | 0.8921 | 0.7202 | 0.7967 | 0.7072 |
| | Neutral | 0.4686 | 0.6328 | 0.5372 | |
| | Positive | 0.5932 | 0.7542 | 0.6637 | |
| Electra | Negative | 0.8282 | 0.6759 | 0.7375 | 0.623 |
| | Neutral | 0.4369 | 0.4882 | 0.4468 | |
| | Positive | 0.398 | 0.5939 | 0.4636 | |
| GAN-BERT | Negative | 0.835 | 0.7752 | 0.7938 | 0.7032 |
| | Neutral | 0.5015 | 0.5484 | 0.5048 | |
| | Positive | 0.6342 | 0.6264 | 0.6253 | |

**Table 6.** Detailed Results for the SemEval Dataset.

| Technique | Class | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|---|
| GAN-BElectra | Negative | 0.4373 | 0.6046 | 0.507 | 0.5741 |
| | Neutral | 0.5992 | 0.5524 | 0.5727 | |
| | Positive | 0.6665 | 0.5857 | 0.6191 | |
| SG-Elect | Negative | 0.4374 | 0.6219 | 0.5116 | 0.5635 |
| | Neutral | 0.5977 | 0.5244 | 0.557 | |
| | Positive | 0.639 | 0.5826 | 0.6055 | |
| Electra | Negative | 0.2835 | 0.3515 | 0.3006 | 0.4536 |
| | Neutral | 0.5288 | 0.596 | 0.5526 | |
| | Positive | 0.5434 | 0.3259 | 0.3445 | |
| GAN-BERT | Negative | 0.4542 | 0.4787 | 0.44 | 0.5208 |
| | Neutral | 0.5736 | 0.5394 | 0.468 | |
| | Positive | 0.634 | 0.5191 | 0.5109 | |

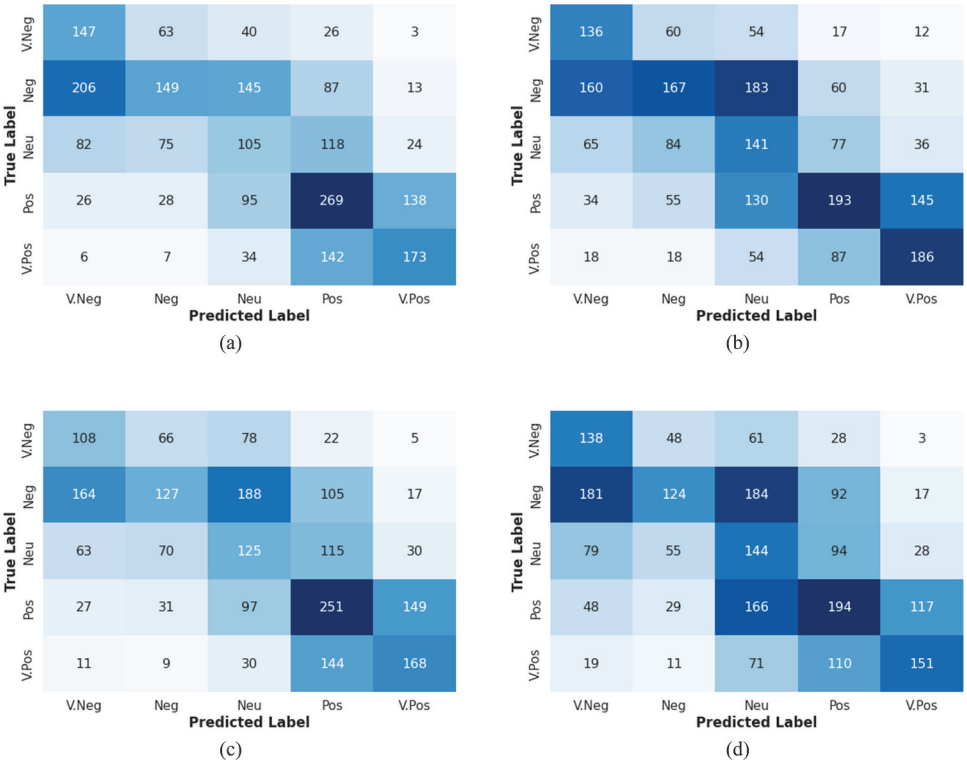


**Figure 9.** Confusion matrices for SST5 dataset for (a) GAN-BElectra (b) SG-Elect (c) Electra and (d) GAN-BERT.

### *SST5 Dataset*

GAN-BElectra outperformed SG-Elect, GAN-BERT, and Electra for the SST5 dataset in mean accuracy. Table 1 shows that GAN-BElectra achieved an average accuracy of 0.3825 while SG-Elect, Electra, and GAN-BERT scored 0.373, 0.3533, and 0.3406, respectively. Table 1 also contains mean F1-Macro scores for all techniques and shows that GAN-BElectra outperforms all other evaluated techniques in this metric as well for the SST5 dataset. Table 4 shows that our proposed solution also performed better than our main baseline SG-Elect and other evaluated techniques in F1-score for individual classes. It outperformed SG-Elect in F1-score for "Very Negative," "Positive," and "Very Positive" class, whereas SG-Elect achieved higher F1-score for "Negative" and "Neutral" class. GAN-BElectra also outperformed GAN-BERT and Electra for F1-score for all individual classes except the "Neutral" class. Confusion matrices in Figure 9 visualize the correct and incorrect predictions by all evaluated techniques for the SST5 dataset.

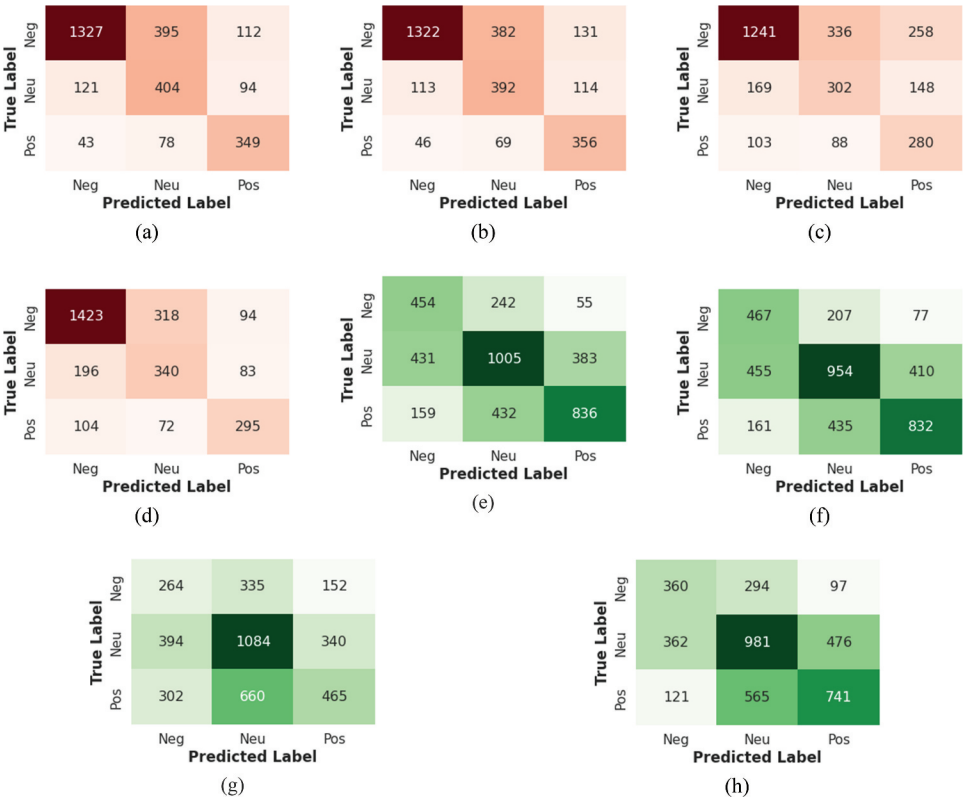


**Figure 10.** Confusion matrices (with **red** shades) for the US Airline dataset for (a) GAN-BElectra (b) SG-Elect (c) Electra and (d) GAN-BART and confusion matrices (with **green** shades) for the SemEval dataset (e) GAN-BElectra (f) SG-Elect (g) Electra and (h) GAN-BART.

## US Airline Dataset

Table 5 shows the detailed comparative classification result for GAN-BElectra, SG-Elect, GAN-BERT and Electra for the US Airline dataset. Our solution achieved an overall accuracy of 0.7108 while SG-Elect, Electra, and GAN-BERT scored 0.7072, 0.623, and 0.7032 respectively. This finding is further strengthened by the fact that GAN-BElectra outperforms all other evaluated techniques in the mean F1-macroscore as well for the US Airline dataset (see Table 1). As demonstrated in Table 5, GAN-BElectra achieved higher F1-score for all three individual classes compared to SG-Elect, GAN-BERT, and Electra. Figure 10 shows the confusion matrices that visualize the correct and incorrect predictions made by all four evaluated techniques for the US Airline dataset.

### SemEval Dataset

Table 6 demonstrates that GAN-BElectra outperforms SG-Elect, GAN-BERT, and Electra in terms of classification accuracy for the SemEval dataset. It achieved an accuracy score of 0.5741 while SG-Elect, Electra, and GAN-BERT achieved 0.5635, 0.5464, and 0.4792 respectively. We also observe in Table 1 that GAN-BElectra outperforms all other evaluated techniques in terms of mean F1-macro score as well for the SemvEval dataset. As it can be seen in Table 6, our solution performed better than SG-Elect in two out of three individual classes, achieving higher F1-score for "Neutral" and "Positive" class while SG-Elect achieved higher F1-score for the "Negative" class. Both Electra and GAN-BERT underperformed in terms of F1-scores for all individual classes compared to SG-Elect and GAN-BElectra. Confusion matrices in Figure 10 visualize the correct and incorrect predictions made by GAN-BElectra, SG-Elect, GAN-BERT, and Electra for the SemEval dataset.

## Ablation

Two major components within GAN-BElectra are GAN-BERT and Electra. As part of our ablation study, we investigated the individual performance of these components on the same test data used for GAN-BElectra and SG-Elect. This analysis provided insight about the effect of individual components that constitute our proposed solution. Table 3, 4, 5, and 6 contain these results for GAN-BERT and Electra. Below we discuss the performance of these individual components on the three datasets used with the proposed GAN-BElectra solution.

### US Airline and SemEval Dataset

GAN-BERT outperformed Electra for our two 3-class sentiment classification datasets (US Airline and SemEval). For US Airline dataset, GAN-BERT achieved an overall accuracy of 0.7032 whereas Electra achieved 0.623. Similarly, for the SemEval dataset, GAN-BERT achieved an overall accuracy score of 0.5208 while Electra achieved 0.4536.

### SST5 Dataset

For SST5 dataset, Electra outperformed GAN-BERT, achieving an overall accuracy of 0.3533 while GAN-BERT scored 0.3406.

## Pseudo Label Generation

We also investigated the accuracy of pseudolabels generated by GAN-BERT which were eventually utilized to train Electra along with the few original labeled data. For three different datasets we have used (SST5, US Airline, and SemEval), the average accuracy of GAN-BERT's generated pseudolabels were 0.3328, 0.7087, and 0.5209 respectively, as demonstrated in Table 3.

In the subsequent section, we discuss the results we achieved with our proposed solution along with the insights gained from the ablation study.

## Discussion

### *Performance Gain with Less Training Steps and Reduced Network Complexity*

Our experiments suggest that the semi-supervised self-training sub-component of SG-Elect (Riyadh and Shafiq 2021) that included a stacked classifier consisting of a Stochastic Gradient Descent (SGD) Classifier ("Optimization: Stochastic Gradient Descent" n.d.) and a XGBClassifier ("XGBoost Documentation – Xgboost 1.5.0-SNAPSHOT Documentation" n. d.) negatively contributed to the overall classification accuracy achieved by SG-Elect. The new technique proposed in this paper, GAN-BElectra, removes that component from the architecture along with SG-Elect's "Combinator" component. This resulted in a performance gain. As we can observe in the reported results, GAN-BElectra performs slightly higher score compared to SG-Elect for the US Airline (GAN-BElectra: 0.7108, SG-Elect: 0.7072), the SemEval (GAN-BElectra: 0.5741, SG-Elect: 0.5635), and the SST5 dataset (GAN-BElectra: 0.3825, SG-Elect: 0.373). While the performance gain is small, the gain is consistent across all evaluated datasets. It is also important to emphasize that compared to SG-Elect, GAN-BElectra reduces the complexity of the architecture as it relieves the need of training an additional machine-learning component that consumes resources and incurs more training cycles.

## Impact of Pseudolabels on the Final Result

GAN-BElectra uses GAN-BERT as the sole pseudolabel generator for the unlabeled training data. As reported in Section VI, we notice that the accuracy of the generated pseudolabels (0.3328, 0.7087, 0.5209 for SST5, US Airline, and SemEval, respectively) resembles the final results of GAN-BElectra for all three datasets (0.3825, 0.7108, 0.5741 in the same order). This highlights the impact of the pseudo label generator, GAN-BERT in our overall architecture, making it an important component of the proposed solution. Without the contribution of GAN-BERT as a pseudolabel generator, the performance of our architecture would have degraded significantly, as can be inferred from our ablation study where we tested the performance of the stand-alone Electra pre-

trained model fine-tuned only on the original few labels. In terms of average accuracy, this fine-tuned stand-alone Electra model achieves 0.3533, 0.6233, and 0.4536 for SST5, US Airline, and SemEval dataset respectively, whereas our proposed model GAN-BElectra demonstrates boost in performance contributed by the pseudolabel generator GAN-BERT, and achieved 0.3825, 0.7108, and 0.5741 in terms of average accuracy for these three datasets in the same order.

## Prediction Trends for Individual Class

### *1) Incorrect Predictions More Likely to Fall into the Adjacent Categories*

We observe that the majority of incorrect predictions typically fall into the adjacent sentiment categories. This means that a "positive" datapoint is more likely to be predicted as "positive" and "neutral" (also "very positive" for 5-class dataset) compared to contrasting labels such as "negative." For instance, for SST5 dataset, out of a total of 556 datapoints with "positive" label, GAN-BElectra correctly predicted the label for 269 datapoints as "positive," and then incorrectly predicted 138 datapoints as "very positive," 95 datapoints as "neutral," only 26 datapoints as "very negative" and 28 as negative. This similar phenomenon has been observed for almost techniques with all datasets (see confusion matrices in Figures 9 and 10).

### 2) Variation in Pseudo Label Accuracy

The accuracy of pseudo label generation by GAN-BERT varied significantly across our three selected datasets. GAN-BERT achieved 0.3328, 0.7087, 0.5209 in pseudo label generation accuracy for SST5, US Airline, and SemEval dataset respectively. Since SST5 is a finer grained sentiment dataset (i.e., 5 sentiment classes whereas the other two datasets have three classes), we expected it to have lower classification accuracy compared to US Airline and SemEval dataset. This is indeed consistent in all other experiments performed in this study.

We also observed that between US Airline and SemEval dataset, though they are both three-class sentiment datasets, GAN-BERT scored 0.7087 for the accuracy for the former compared to 0.5209 for the latter. We believe this is due to the different composition of the two datasets. For example, US Airline has a significantly higher number of "negative" datapoints compared to the other two classes. On the other hand, SemEval contains much less "negative" datapoints compared to the other two classes. However, Figure 7 and Figure 8 show that this class imbalance in the SemEval dataset is visibly less significant compared to US Airline dataset. Confusion matrices in Figure 10 for test data show that all evaluated techniques performed more accurate predictions for

the neutral class for the SemEval dataset compared to the other two classes. For US Airline dataset, it is the negative sentiment class where more accurate predictions occurred across all techniques. Based on these observations, we argue that the different compositional attributes of these two datasets (e.g., significantly higher number of "negative" datapoints in the US Airline dataset compared to the SemEval dataset which was visibly less imbalanced) might have an impact on the overall differences in the accuracy of pseudolabel generation for these two datasets. As mentioned in Section IV, our rationale for not artificially balancing the composition of the datasets was due to the fact that the primary training data used in our experiments were organically balanced since we used 50 datapoints for each class for each dataset in each of our experiments as the "few original labeled training data." In addition, along with accuracy, we also reported F1-macro score which provided more reliable comparison of techniques when class imbalance is present in the selected dataset. However, we believe that this may still have an impact on the pseudolabel generation accuracy as well as the variations in the number of correctly predicted labels for our test data (see Confusion matrices in Figure 9 and 10). We recognize that this warrants further investigation.

It is also noteworthy that between GAN-BERT and Electra, the two main components of GAN-BElectra, GAN-BERT outperforms Electra for two 3-class sentiment datasets, while Electra outperforms GAN-BERT for SST5 dataset which contains 5-class sentiment data. Fine-tuning stand-alone Electra is significantly faster than fine-tuning GAN-BERT since GAN-BERT uses a semi-supervised training method, which essentially performs many iterations of supervised training with varying training data. Despite this, Electra achieving higher accuracy for SST5 than GAN-BERT is an interesting observation which warrants further investigation.

### Difference in Accuracy across Datasets

We observed that GAN-BElectra's performance expectedly degrades with increase in the number of sentiment classes. However, it is of note that despite having identical sentiment categories, the two 3-class sentiment datasets used in our experiments yielded significantly different results for all architectures tested, including the proposed GAN-BElectra. Similar to our primary baseline (SG-Elect) and other two evaluated techniques (GAN-BERT, Electra), GAN-BElectea achieved a significantly higher classification accuracy for US Airline dataset (0.7108) compared to SemEval dataset (0.5741). As we argued above for the same phenomenon for the pseudolabel generation by GAN-BERT, we believe this may be due the differences in the data composition across these two datasets. However, we believe it is worth investigating further in future studies.

### Impact of Double fine-tuning

We observe that double fine-tuning Electra, once with pseudolabels and next with a few original labels boosts accuracy of this transformer network compared to training it only once with either pseudolabels or the original few labels. This confirms the finding in previous studies (Bowon and Jin Choi 2020) where researchers noticed similar improvement in classification accuracy with fine-tuning pretrained models twice.

## Conclusion

In this research, we focus on the problem of lack of labeled data in multi-class sentiment analysis task. We aimed to build a sentiment analysis technique that performs well with a few labeled data. We propose a novel technique for multi-class sentiment analysis named GAN-BElectra, which builds upon its predecessor and baseline technique named SG-Elect. GAN-BElectra achieves a higher classification accuracy compared to SG-Elect with reduced architecture complexity. GAN-BElectra's architecture leverages GAN-BERT (Croce, Castellucci, and Basili 2020) as its sole pseudolabel generator, and Electra (Clark et al. 2020) as the final classifier. This final component is fine-tuned twice; first with the pseudo-labels generated by GAN-BERT, and next with the few original labeled data. This overall architecture eventually predicted sentiment with more accuracy compared to its primary baseline SG-Elect for all three datasets we experimented with.

Although GAN-BElectra demonstrated better performance compared to its predecessor (i.e., SG-Elect), the margin of improvement is not exceptionally high. This suggests there is still room for further improvement in this area. Another noteworthy limitation of the proposed solution is that it requires a multi-step training process involving training the GAN-BERT component first and then training the Electra component. Achieving an end-to-end solution without requiring multi-step training process is a worthwhile future exploration. Finally, it would be interesting to explore the efficacy of GAN-BElectra and its potential future variants with more NLP classification tasks other than sentiment analysis.

## Acknowledgments

The authors acknowledge support from Natural Sciences and Engineering Research Council of Canada (NSERC) and Carleton University, Canada. The work presented in this paper is built upon and extended from the work presented in the earlier paper (Riyadh and Shafiq 2021).

## Disclosure Statement

No potential conflict of interest was reported by the author(s). The work presented in this paper is built upon and extended from the work presented in the earlier paper by the same authors (Riyadh and Shafiq 2021).

## Funding

This work is funded by Natural Sciences and Engineering Research Council of Canada (NSERC) and Carleton University, Canada.

## References

Abonizio, H. Q., E. Cabrera Paraiso, and S. Barbon Junior. 2021. "Toward text data augmentation for sentiment analysis IEEE Transactions on Artifical Intelligence ." , , 1–1 2691-4581 . 10.1109/tai.2021.3114390. Early Access

Ashish, V., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. "Attention is all you need." In *Advances in Neural Information Processing Systems*, 2017 December:5999–6009. Neural information processing systems foundation. https://arxiv.org/abs/1706.03762v5.

Barriere, V., and A. Balahur. 2020. "Improving sentiment analysis over non-english tweets using multilingual transformers and automatic translation for data-augmentation," October. http://arxiv.org/abs/2010.03486.

Bowon, K., and H. Jin Choi. 2020. Twice fine-tuning deep neural networks for paraphrase identification. *Electronics Letters* 56 (9):449–50. doi:10.1049/el.2019.4183.

Brownlee, J. 2019 Statistical methods for machine learning. Discover how to transform data into knowledge with Python (Machine Learning Mastery) 291 Accessed 10 April 2022 https://machinelearningmastery.com/statistics_for_machine_learning . .

Clark, K., M.-T. Luong, Q. V. le, and C. D. Manning. 2020. "ELECTRA: pre-training text encoders as discriminators rather than generators." In *ICLR*.

"Cloud tensor processing units (TPUs) | google cloud." n.d. Accessed May 2, 2022. https://cloud.google.com/tpu/docs/tpus.

"Colaboratory – Google." n.d. Accessed May 15, 2021. https://research.google.com/colaboratory/faq.html.

Croce, D., G. Castellucci, and R. 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* July 2020 Online, 2114–19. Association for Computational Linguistics.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1 (October):4171–86.

Edunov, S., M. Ott, M. Auli, and D. Grangier. 2018. Understanding back-translation at scale EMNLP October 31–November 4 (Association for Computational Linguistics)489–500 Brussels, Belgium.

Edwards, A., A. Ushio, J. Camacho-Collados, H. de Ribaupierre, and A. Preece. 2021. "Guiding generative language models for data augmentation in few-shot text classification," November. http://arxiv.org/abs/2111.09064.

Fan, M., D. Meng, Q. Xie, L. Zina, and X. Dong. 2017 Self-Paced Co-Training International Conference on Machine Learning August 6-11, 2017 Sydney, Australia . . PMLR 2275–2284 .

"GitHub - Crux82/Ganbert: Enhancing the BERT training with semi-supervised generative adversarial networks." n.d. Accessed May 26, 2021. https://github.com/crux82/ganbert.

Hemmatian, F., and M. Karim Sohrabi. 2019. A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review* 52 (3):1495–545. doi:10.1007/s10462-017-9599-6.

Kim, H., J. Son, and Y.-S. Han. 2022. "LST: Lexicon-guided self-training for few-shot text classification," February. http://arxiv.org/abs/2202.02566.

Kingma, D. P., and B. Jimmy Lei 2015. "Adam: A method for stochastic optimization." In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* May 7 - 9, 2015 San Diego, CA, USA, International Conference on Learning Representations. ICLR.

Kumar, M. P., B. Packer, and D. Koller. 2010. Self-paced learning for latent variable models. *Nips* 1:2.

Liddy, E. D. 2001. "Natural language processing." https://surface.syr.edu/istpub.

Liu, S., X. Cheng, L. Fuxin, and L. Fangtao. 2015. TASC: Topic-adaptive sentiment classification on dynamic tweets. *IEEE Transactions on Knowledge and Data Engineering* 27 (6):1696–709. doi:10.1109/TKDE.2014.2382600.

Loshchilov, I., and F. Hutter. 2019. "Decoupled weight Decay Regularization." In *ICLR*.

Mastoropoulou, E.-P. 2019. Enhancing Deep Active Learning Using Selective Self-Training For Image Classification (KTH ROYAL INSTITUTE OF TECHNOLOGY)Accessed 10 April 2022 https://www.diva-portal.org/smash/get/diva2:1414329/FULLTEXT01.pdf .

"Matplotlib: Python plotting — matplotlib 3.4.2 documentation." n.d. Accessed May 15, 2021. https://matplotlib.org/.

Mazharul Islam, S. M., X. Dong, and G. de Melo. 2020. "Domain-specific sentiment lexicons induced from labeled documents." In *Proceedings of the 28th International Conference on Computational Linguistics* Barcelona, Spain (Online) (International Committee on Computational Linguistics), 6576–87. . http://sentimentanalysis.org.

Miao, Z., L. Yuliang, X. Wang, and W.-C. Tan. 2020. Snippext: semi-supervised opinion mining with augmented data In Proceedings of The Web Conference 2020 20-24 April 2020 617–628 Taipei, Taiwan. doi:10.1145/3366423.3380144.

Mitchell, T. M. 1997. *Machine Learning*. New York: McGraw-Hill International.

Ning, L., C.-Y. Chow, and J.-D. Zhang. 2020. SEML: A semi-supervised multi-task learning framework for aspect-based sentiment analysis. *IEEE Access* 8 (October):189287–97. doi:10.1109/access.2020.3031665.

"NumPy." n.d. Accessed May 15, 2021. https://numpy.org/.

"Numpy.argmax — numpy v1.22 manual." n.d. Accessed May 2, 2022. https://numpy.org/doc/stable/reference/generated/numpy.argmax.html.

"Optimization: Stochastic gradient descent." n.d. Accessed May 2, 2022. http://deeplearning.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/.

Patel, D. . Approaches for sentiment analysis on twitter: A STATE-OF-ART STUDY Accessed 10 April 2022 https://arxiv.org/abs/1512.01043 doi:https://doi.org/10.48550/arXiv.1512.01043 arXiv:1512.01043 .

Radford, A., W. Jeffrey, R. Child, D. Luan, D. Amodei, and I. Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1 (8):9. https://github.com/codelucas/newspaper.

Ricci, F., B. Shapira, and L. Rokach. 2015. Recommender systems: Introduction and challenges. In *Recommender Systems Handbook*, *2nd* 1 (Boston, MA: Springer) 1–34 ed. doi:10.1007/978-1-4899-7637-6_1.

Riyadh, M., and M. Shafiq. 2021. "Towards multiclass sentiment analysis with limited labeled data." In *IEEE International Conference on Big Data* December 15-18, 2022 Online, 4955–64.

Rosenthal, S., N. Farra, and P. Nakov. 2017. "SemEval-2017 task 4: sentiment analysis in twitter." In *Proceedings of the 11th International Workshop on Semantic Evaluation ({S}em {E}val-2017)*, 502–18. Vancouver, Canada: Association for Computational Linguistics. 10.18653/v1/S17-2088.

"Scipy.special.softmax — sciPy v1.8.0 manual." n.d. Accessed May 2, 2022. https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.softmax.html.

"Self training in semi-supervised learning — scikit-learn documentation." n.d. Accessed November 15, 2021. https://scikit-learn.org/stable/modules/semi_supervised.html#self-training.

Socher, R., A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. 2013. "Recursive deep models for semantic compositionality over a sentiment treebank." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* 18-21 October 2013 Seattle, Washington, USA, 1631–42. Association for Computational Linguistics. http://nlp.stanford.edu/.

Sun, Y., S. Wang, L. Yukun, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and W. Hua 2019. "ERNIE: Enhanced Representation Through Knowledge Integration." In *ArXiv E-Prints*. https://github.com/PaddlePaddle/.

"TensorFlow." n.d. Accessed May 15, 2021. https://www.tensorflow.org/.

"TensorFlow hub - electra large." n.d. Accessed May 26, 2021. https://tfhub.dev/google/electra_large/2.

"Twitter US airline sentiment | kaggle." n.d. Accessed May 22, 2021. https://www.kaggle.com/crowdflower/twitter-airline-sentiment.

Vries, P. G. D. 1986. Stratified random sampling. In *Sampling theory for forest inventory*, 31–55 doi:10.1007/978-3-642-71581-5_2. Berlin, Heidelberg: Springer.

"What is a GPU? graphics processing units defined." n.d. Accessed May 15, 2021. https://www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, et al. 2019. Transformers: state-of-the-art natural language processing Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations 2020 (Association for Computational Linguistics)38–45 Online. . doi:10.18653/v1/2020.emnlp-demos.6.

"XGBoost documentation — xgboost 1.5.0-SNAPSHOT documentation." n.d. Accessed May 16, 2021. https://xgboost.readthedocs.io/en/latest/.

Xinzhe, L., Q. Sun, Y. Liu, S. Zheng, Q. Zhou, T.-S. Chua, and B. Schiele. 2019. Learning to self-train for semi-supervised few-shot classification. *Advances in Neural Information Processing Systems* 32:10276–86.

Yang, T., H. Linmei, C. Shi, J. Houye, L. Xiaoli, and L. Nie. 2021. HGAT: Heterogeneous Graph attention networks for semi-supervised short text classification. *ACM Transactions on Information Systems* 39:3. doi:10.1145/3450352.

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. *Association for Computational Linguistics (ACL)* 189–96. doi:10.3115/981658.981684.

Zhang, L., S. Wang, and B. Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8 (4):e1253.