Knowledge-Based Systems 284 (2024) 111291

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

CDGAN-BERT: Adversarial constraint and diversity discriminator for semi-supervised text classification

Nai Zhou [a], Nianmin Yao [a,d,*], Nannan Hu [c], Jian Zhao [b,d], Yanan Zhang [e]

[a] School of Computer Science, Dalian University of Technology, Dalian, 116024, China
[b] School of Automotive Engineering, Dalian University of Technology, Dalian, 116024, China
[c] School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 10000, China
[d] Ningbo Institute of Dalian University of Technology, Ningbo, 315016, China
[e] Automotive Data of China Co. Ltd, Tianjin, 300300, China

ARTICLE INFO

Keywords:
Semi-supervised generative adversarial learning
The adversarial constraint
The diversity discriminator
Text classification

ABSTRACT

Semi-supervised generative adversarial learning (SS-GAN) significantly improves the model's performance with limited labeled data, especially on text classification. However, these SS-GAN-like works pay more attention to the more real properties of the generated samples and ignore their adversarial nature, which may prevent the model from learning more higher-quality data representations. And the existing single discriminator may produce discrimination ambiguity when discriminating against tasks with different discriminative properties (simultaneously identifying real or fake and classifying categories), which may harm the model's classification performance. In this paper, we propose a novel CDGAN-BERT, an SS-GAN-based architecture with the adversarial constraint and a diversity discriminator. Specifically, CDGAN-BERT first focuses on adversarial constraint, which calculates the model's intrinsic state representation with the reciprocal of Mean Squared Error (MSE), further keeping adversarial of the generated samples relative to the real data distribution. Then, a diversity discriminator is designed to improve the model's classification performance by alleviating discriminative ambiguity, which refines tasks with different discriminative attributes by adding an additional authenticity discriminator. We validate our model on 6 text datasets and achieve significant improvements, especially with limited supervision. The experimental results show that our method outperforms or gets comparable results to other state-of-the-art semi-supervised learning methods on several datasets. Especially for the datasets of QC-Fine and QC-Coarse with limited labeled data, our CDGAN-BERT achieves the best average Micro-F1 scores of 72.5% and 92.037%, respectively.

1. Introduction

In recent years, Semi-Supervised learning (SSL) has achieved excellent performance without relying on a large amount of labeled data, which has attracted increasing attention in NLP and CV [1–4], especially on text classification tasks [4–10]. Although there are a lot of works on semi-supervised text classification, many of them rely heavily on additional auxiliary language models or techniques, such as data augmentation or back-translation [7,8,11]. Implementing these tasks would need a lot of time and processing power in the real world. In order to overcome these challenges, recent research has effectively coupled semi-supervised learning with generative adversarial networks, yielding outstanding results in a variety of tasks in the CV and NLP domains. Taking inspiration from these advancements, in this paper, we adopt the SS-GAN approach for semi-supervised text classification. However, among SS-GAN-based works [4,10,12–15], they usually ignore the following issues. One is that research usually imposes more constraints on the generator to generate a more realistic data distribution, thereby ignoring the 'adversarial' in adversarial learning, which may prevent the model from learning higher-quality data representations. And the other is the discriminative ambiguity caused by tasks with different discriminative properties. In other words, the existing single discriminator needs to distinguish real or fake and classify categories simultaneously, which may affect the classification performance of the model.

In this paper, we address the issues mentioned above by introducing the adversarial constraint and a diversity discriminator into the SS-GAN-based model. Inspired by 'adversarial' in Generative Adversarial Networks (GAN) [16], we introduce the adversarial constraint to keep

* Corresponding author at: School of Computer Science, Dalian University of Technology, Dalian, 116024, China.
E-mail addresses: zhounai@mail.dlut.edu.cn (N. Zhou), lucos@dlut.edu.cn (N. Yao), hunan246@bupt.edu.cn (N. Hu), jzhao@dlut.edu.cn (J. Zhao), zhangyanan@catarc.ac.cn (Y. Zhang).

https://doi.org/10.1016/j.knosys.2023.111291
Received 19 January 2023; Received in revised form 27 September 2023; Accepted 8 December 2023
Available online 9 December 2023

adversarial of the generated samples. Most works usually constrain the generator to pursue better generative representation in GAN-based training. However, after a certain period of training, extremely realistically generated data may not encourage the model to continue learning higher-quality data representations. In particular, the optimization of the discriminator may become ineffective. The absence of adversarial learning between the generator and the discriminator, assuming that the produced samples are replicas of the original samples, is not helpful for optimizing model training. Back to the essence of 'adversarial' in GAN, intuitively, the samples generated by the generator should approximate the real examples and remain adversarial. We hope that the samples generated by the generator have rich diversity while ensuring the correctness. These adversarial samples can continuously fine-tune BERT to improve its presentation ability. Therefore, we introduce adversarial constraints by performing the reciprocal of MSE on generated and real data distributional representations. The adversarial constraint emphasizes that the generated data needs to remain adversarial, which will help the model learn higher-quality data representations in the 'adversarial'. In other words, comparing with previous works paid more attention to the more realistic samples generated by the generator. The aim of our work is to improve the classification performance of the model by enhancing the high-quality representation ability of BERT. We propose to directly impose adversarial constraints on the distribution of generated samples to make the generated samples have 'adversarial' (richer diversity) while maintaining reality. These 'adversarial' examples can promote BERT to achieve higher quality representations. We believe that the adversarial samples in the generative adversarial learning process not only need to be more real and similar, but also need to have an impact on model training.

Besides, the SS-GAN-based model shared a single discriminator for classification (classify categories) and identification (identify real or fake). A single discriminator will cause discriminative ambiguity, which may degrade the model's overall performance. Most existing SS-GAN-like structure works adopt a discriminator design of $k+1$ categories (k represents the number of real sample categories) [10,12,15,17,18]. The purpose is to classify the real samples into one of the k classes and divide the generated fake samples into class $k+1$. In other words, a single discriminator needs to classify categories while identifying real or fake samples. However, identification of real or fake and category classification are two tasks with different discriminative properties. It is challenging for a single discriminator to complete these two tasks simultaneously, and discriminating ambiguity are prone to occur. Therefore, we design a diversity discriminator by adding an additional authenticity discriminator based on the original discriminator. The authenticity discriminator is actually a real and fake binary classifier, which can better learn the differences between real samples and fake samples, and more easily identify the authenticity of samples. By introducing the above ability to distinguish real or fake, the discriminative ambiguity in the single discriminator is alleviated. Furthermore, the improvement of the whole model identification ability could help the generator improve its generation ability.

To sum up, we propose a novel CDGAN-BERT, which expends the SS-GAN-based architecture with the adversarial constraint and a diversity discriminator. The main purpose of our proposed method is to achieve effective text classification using a small amount of labeled and a large amount of unlabeled data without relying on additional models and data augmentation techniques, such as data augmentation achieved through the use of additional trained translation system models. The CDGAN-BERT has the following advantages: It reduces the cost of additional computing resources by not using data augmentation techniques and other support with task-specific models; It further enables the model to learn higher-quality data representations and improves its performance. CDGAN-BERT does not rely on additional models for data augmentation (such as Back-Translation and Interpolation-Based data augmentation). We believe and demonstrate experimentally that CDGAN-BERT provides a new solution for the classification task with only a few annotated samples during model training. Our contributions can be summarized as follows:

- We propose a novel CDGAN-BERT method, which combines the SS-GAN-like architecture and BERT, and builds an adversarial constraint module and a diversity discriminator module on this basis.
- The adversarial constraint module enables the generator to generate adversarial samples that are both realistic and remain adversarial by constraining the intrinsic state representation, further enabling BERT to learn to a higher representation capability. To the best of our knowledge, this is the first work to introduce the adversarial constraint on the model's intrinsic state representation into an SS-GAN-like architecture, especially in the semi-supervised text classification.
- The diversity discriminator module consists of an authenticity discriminator and a classification discriminator, which can alleviate the discrimination ambiguity of a single discriminator under a multi-attribute task, and thus improve the model's authenticity discrimination ability and classification performance.
- We conduct experiments on 6 text datasets. The experimental results demonstrate that our model (CDGAN-BERT) outperforms or gets comparable results to other state-of-the-art methods on semi-supervised text classification, notably more significant improvements with extremely limited supervision.

The rest of this article is organized as follows: we introduce some related works about semi-supervised generative adversarial learning in Section 2. The details about our proposed methodology are presented in Section 3. Next, we evaluate our method on six text classification datasets and the related ablation experiments, respectively. For all experimental results, we report that in Section 4. Finally, we conclude our work in Section 5.

2. Related works

2.1. Semi-supervised text classification

Recently, semi-supervised learning has received extensive attention in the NLP community, especially on text classification [1,5,9,19,20]. For instance, some researches utilize variational auto encoders (VAEs) to reconstruct sentences and predict sentence labels from latent variables learned from the reconstructions, building sequence-to-sequence classification model [5,21,22]. Some works used adversarial training to achieve supervised and semi-supervised text classification, such as AAE [23] and Cross-lingual transfer [24]. There are also some works use virtual adversarial training to achieve semi-supervised text classification by adding perturbation to word vectors [1,6]. In addition, many studies first increase the amount of labeled data through techniques such as data augmentation or back-translation [25] and then train classification model, e.g., TMix [8] achieved data expansion by performing interpolation operations in the hidden space representation of text samples. UDA [11] and SMDA [26] adopt back translation [25] and word replacement on unlabeled data. FLiText [7] leveraged consistency regularization and data augmentation techniques for semi-supervised text classification. S2TC-BBD [9] got a good score on semi-supervised text classification by using Balanced Deep Representation Distributions to get more confident pseudo-labels on unlabeled text. Unlike these works, our method does not need to increase the amount of labeled data through some techniques and additional trained models, such as back-translation, interpolation-based data augmentation, and the trained translation models.

2.2. Semi-supervised generative adversarial networks

There are many works applied SS-GAN for image processing, and that architecture have been shown to be effective [12–14,27–31]. For example, [12] proposed SS-GAN architecture with a variety of new architectural features for semi-supervised image classification by improving the GAN [16]. [32] proposed CR-GAN by adding consistency

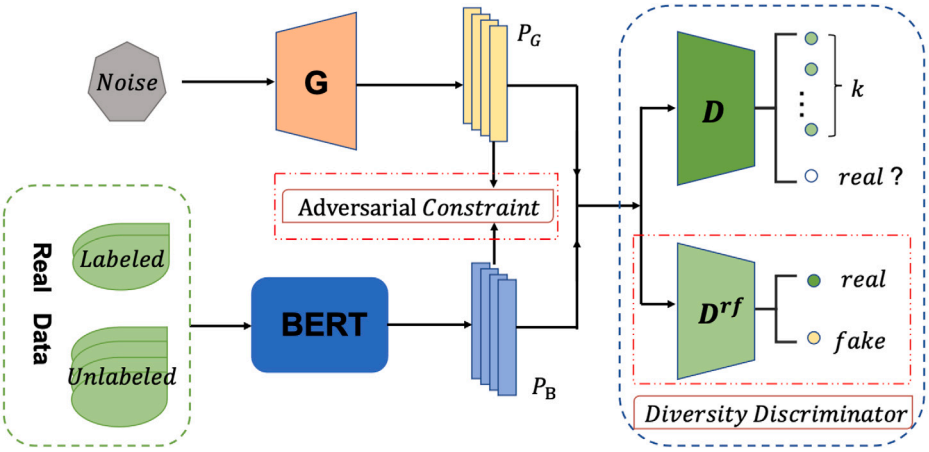

Fig. 1. The overall framework of our proposed CDGAN-BERT. The random noise signal is fed into the generator to obtain a fake data distribution representation (P_G). P_B is the pre-trained BERT's output of the real samples. We perform adversarial constraint for P_G and P_B. The diversity discriminator consists of D and D^{rf}. The output of D is a $k+1$ vector of *logits*, and the task of D^{rf} is to distinguish whether a sample is real or fake.

regularization to the discriminator and penalizing the sensitivity of the discriminator to the random augmentations. EC-GAN [14] implemented the training of the SS-GAN architecture by applying two discriminators with different functions and achieved good results on low-sample image classification. However, SS-GAN in semi-supervised text classification has not been widely applied. For instance, Kernel-based GAN [33] did a sentence classification task by applying a Kernel-based Deep Architecture in an SS-GAN perspective. [10] proposed GAN-BERT, which has achieved competitive results with the limited labeled text through extending the BERT by using SS-GANs in generative adversarial learning. Of the aforementioned work, our method combines the BERT-like architecture with the semi-supervised adversarial generation architecture. And based on that, we constrain the intrinsic representation of the model's hidden layers to further improve the quality of the generated data representation, and design a diversity discriminator for classification tasks with different attributes to improve the model's identify and classification abilities.

3. Methodology

CDGAN-BERT is a novel model for Text Classification with limited annotated samples. Its main novelty comes from two parts, the adversarial constraint of the model's intrinsic state representation and the use of this constraint in combination with the diversity discriminator for model's adversarial training. These two improvements are shown in the red dashed part in Fig. 1. Before elaborating on the details, we first introduce some specific representations below.

Consider that there is limited labeled data $\mathrm{D}_L = \{x_i, y_i\}_{i=1}^{N_L}$ and a large amount of unlabeled data $\mathrm{D}_U = \{x_i\}_{i=1}^{N_U}$, where x_i denotes a text sequence sample, and y_i is its corresponding label, and $y_i \in \{1, \ldots, k\}$ indicating that there are k categories. In addition, N_L and N_U represent the number of labeled and unlabeled data, respectively, and N_U is much larger than N_L ($N_U \gg N_L$). The key notations and their descriptions are listed in Table 1.

3.1. Overview of CDGAN-BERT

The overall framework of our proposed CDGAN-BERT is shown in Fig. 1. The goal of the text classification is to learn a mapping function $\theta : x \rightarrow y$ with limited annotated text data and a large amount of unlabeled text data. In CDGAN-BERT, we perform adversarial constraint module computation on the outputs of the Generator (G) and the Encoder ($BERT$) with the reciprocal of MSE. In addition, we design a diversity discriminator module, which includes D to classify different $k+1$ categories and D^{rf} to distinguish between real and fake samples. Finally, we apply the adversarial constraint and the diversity discriminator to model training. Next, we will introduce more specific details of the CDGAN-BERT.

3.2. Adversarial constraint

The Adversarial Constraint on the model's intrinsic state representation aims to keep adversarial of the samples generated by the generator. In other words, generated samples are more similar to the real sample distribution while maintaining a certain difference from the real sample distribution. Keeping the generated samples adversarial will help the model learn more feature information from 'adversarial'. In this study, we constrain the intrinsic representation of the model's hidden layers with the reciprocal of MSE. We directly perform adversarial constraint on the output of G and the output of $BERT$, and two outputs are denoted by P_G and P_B, respectively. So, our new $Adversarial\ Constraint\ Calculation\ loss$ objective for G is denoted as $L_{G_{CoCa}}$. The formula is as follows:

$$L_{G_{CoCa}} = \left\{ \left\| \mathbb{E}_{noise \sim G}\ P_G - \mathbb{E}_{x \sim BERT}\ P_B \right\|_2^2 \right\}^{-1} \tag{1}$$

where the P_G refers to the output produced by $noise$ passing through G, the P_B refers to the output produced by real data passing through $BERT$.

3.3. Diversity discriminator

Intuitively, it is very difficult and unnatural for a single discriminator to solve tasks with different properties simultaneously. To improve the model's ability to distinguish tasks with different attributes and ameliorate the unnaturalness of the original discriminator. We design a diversity discriminator by adding an auxiliary discriminator that identifies real or fake. As shown in the blue dotted line in Fig. 1, the Diversity Discriminator consists of two different Classifiers: one is D, which is applied to judge the sample's authenticity and category; the other is D^{rf}, which only considers whether it is a real sample or a fake sample.

The two discriminators in the CDGAN-BERT architecture will perform different tasks separately. More specifically, the task of D is expressed as: $p(\hat{y} = y \mid x, y \in \{1, \ldots, k, k+1\}; \boldsymbol{\theta})$, which means that samples are classify into a certain category in $k+1$. At the same time, the D^{rf} distinguishes whether the sample is real or fake, and the formula is defined as: $p(\hat{y} = y \mid x, y \in \{real,\ fake\}; \boldsymbol{\theta})$. Where $\boldsymbol{\theta}$ is the model parameters.

Table 1
Key notations and descriptions.

Notation	Description	Notation	Description
x_i	The text sample	$L_{D_{supervised}}$	The standard Cross-Entropy loss
y_i	The class label of the text sample	$L_{D^f_{unsupervised}}$	The error of not identifying fake examples
k	The number of categories	$L_{D^{rf}_{unsupervised}}$	The loss of D^{rf}
N_L	The number of labeled samples	$L^{D^{rf}}_{G_{unsupervised}}$	the error that fake examples are correctly identified by D^{rf}
N_U	The number of unlabeled samples	$L_{D^r_{unsupervised}}$	The error of incorrectly identifying real examples as fake
θ	The model parameters	P_B	The distribution representation of $BERT$ output
D	The $(k+1)$-class classifier	P_G	The distribution representation of G output
G	The Generator	$L_{G_{CoCa}}$	The Adversarial Constraint Calculation loss
$BERT$	The Encoder	$L_{G_{feature}}$	The feature matching loss of G
L_D	The loss of D^r	$L_{D_{unsupervised}}$	Wrongly identify real unlabeled samples as fake samples; not identify fake samples
L_G	The loss of G	$L^D_{G_{unsupervised}}$	The error that fake examples are correctly identified by D
D^{rf}	The authenticity discriminator	$noise$	A 100-dimensional noise vector sampled from the Normal Distribution $\mathcal{N}(0,1)$

3.4. CDGAN-BERT learning

Consider a standard classifier to classify data x into one of K possible categories and get a K-dimensional $logits$ output vector $l_1, \dots, l_K$. Then, the category probability after $softmax$ is denoted as $p(y = j|x,\theta) = \frac{exp(l_j)}{\sum_{k=1}^{K} exp(l_k)}$. To train the CDGAN-BERT k-class classifier, the objective of the Diversity Discriminator is expressed as follows. $p(\hat{y} = y\ |x, y = k+1;\theta)$ denotes the probability that a generic example x is associated with the fake class, provided by the model θ. $p(\hat{y} = y\ |x, y \in \{1,\dots,k\};\theta)$ represents the probability that x is considered a real sample and therefore belongs to one of the k classes. $p(\hat{y} : real|x;\ \theta)$ and $p(\hat{y} : fake|x;\ \theta)$ indicate that x is considered a real sample and a fake sample, respectively.

So, in the CDGAN-BERT training, the loss calculation formula of the Diversity Discriminator is defined as: For D^{rf}:

$$\begin{aligned} L_{D^{rf}_{unsupervised}} &= L_{D^r_{unsupervised}} + L_{D^f_{unsupervised}} \\ &= -\mathbb{E}_{(x)\sim P_B} \log[1 - p(\hat{y} : fake|x;\theta)] \\ &\quad - \mathbb{E}_{(x)\sim P_G} \log[p(\hat{y} : fake|x;\theta)] \end{aligned} \tag{2}$$

that $L_{D^{rf}_{unsupervised}}$ computes the error of incorrectly identifying real examples as fake and not identifying fake examples. For D:

$$L_D = L_{D_{supervised}} + L_{D_{unsupervised}} + \beta * L_{D^{rf}_{unsupervised}} \tag{3}$$

where:

$$L_{D_{supervised}} = -\mathbb{E}_{(x,y)\sim P_B} \log[p(\hat{y} = y|x, y \in \{1,\dots,k\};\theta)] \tag{4}$$

$$\begin{aligned} L_{D_{unsupervised}} &= -\mathbb{E}_{(x)\sim P_B} \log[1 - p(\hat{y} = y\ |x, y = k+1;\theta)] \\ &\quad - \mathbb{E}_{(x)\sim P_G} \log[p(\hat{y} = y\ |x, y = k+1;\theta)] \end{aligned} \tag{5}$$

In the above formula, $L_{D_{supervised}}$ is the standard cross-entropy loss, which measures the error of misclassification of labeled real data. The unsupervised loss $L_{D_{unsupervised}}$ is produced by D, the function is similar to $L_{D^{rf}_{unsupervised}}$, contains two aspects: wrongly identifies real unlabeled samples as fake samples and does not identify fake samples.

At same time, the loss calculation formula of the G is defined as:

$$L_G = L^D_{G_{unsupervised}} + L_{G_{feature}} + \gamma * L^{D^{rf}}_{G_{unsupervised}} + \tau * L_{G_{CoCa}} \tag{6}$$

where:

$$L^D_{G_{unsupervised}} = -\mathbb{E}_{(x)\sim P_G} \log[1 - p(\hat{y} = y\ |x, y = k+1;\theta)] \tag{7}$$

$$L_{G_{feature}} = \left\| \mathbb{E}_{x\sim P_B}\ f(x) - \mathbb{E}_{x\sim G}\ f(x) \right\|_2^2 \tag{8}$$

$$L^{D^{rf}}_{G_{unsupervised}} = -\mathbb{E}_{(x)\sim P_G} \log[1 - p(\hat{y} : fake|x;\theta)] \tag{9}$$

$L_{G_{CoCa}}$ is as described in Eq. (1). $L^D_{G_{unsupervised}}$ and $L^{D^{rf}}_{G_{unsupervised}}$ measure the error that fake examples are correctly identified by D and D^{rf}, respectively. As used in the SS-GAN [12], $L_{G_{feature}}$ is similar calculation on the output of the activation on an intermediate layer $f(x)$ of D to optimize G.

3.5. Training and inference

In the CDGAN-BERT framework, the diversity discriminator is applied to classification and identification, meanwhile a generator G acting adversarially. In particular, the diversity discriminator is composed of D and D^{rf}, which are Multi-Layer Perceptron (MLP) structures, and the last layer is a $Softmax$ activated layer. The difference is that the output of D and D^{rf} are $(k+1)$-dimensional and 2-dimensional logits vector. The G is another MLP that takes in input a 100-dimensional noise vector drawn from the Normal Distribution $\mathcal{N}(0,1)$, and then produces $p_G \in \mathbb{R}^{\mathbf{d}}, d = 768$. In the forward step of CDGAN-BERT, when an example is sampled from real data $(D_L \cup D_U)$, D should classify it into one of the k classes, while D^{rf} should identify it as a real sample. For samples generated by G, D should classify it as class $k+1$; and D^{rf} judge it as fake. As described in above section, the training process of our proposed model architecture is to optimize three competing losses: $L_{D^{rf}}$, L_D and L_G.

During the back-propagation stage, the supervised loss $L_{D_{supervised}}$ is calculated only from the labeled examples. For the unlabeled examples, when they are misclassified as $k+1$ category, they will be considered to the $L_{D_{unsupervised}}$ loss calculation; when they are classify into one of k categories, their contribution to the loss calculation will be discarded. And, all real examples will be considered to compute the $L_{D^{rf}}$ loss. Besides, the examples P_G generated by G contribute to $L_{D^{rf}}$, L_D and L_G. $L_{D^{rf}}$ and L_D adopt different discriminate strategies for different tasks, and they reflect punishment from different sources when the examples generated by G are not recognized. And vice-versa for G. Meanwhile, the parameters of the pre-trained BERT will be updated when the diversity discriminator updating. In the whole training process, our proposed model only adopts limited labeled data and many unlabeled data. Other than that, we did not perform any data augmentation operations on the labeled data. After training, we discard G and D^{rf}, leaving only the remaining architectures for inference. This also means no additional cost in the inference phase compared with the standard BERT.

Furthermore, we provide the computational complexity matrix of our proposed method. Let n be the length of the sequence, the dimension of the random noise of the generator input is $d_1 \times n$, and the dimension of the output data is $d_2 \times n$. The generator G, classifier D and authenticity discriminator D^{rf} all use L-layer Perceptron, where the latter two also add an activation function layer. We use a BERT-based model, where the number of hidden layers is $L_B = 12$, hidden_size is 768, and $d = 64$ and $h = 12$ denote the head_size and the number of head, respectively. The formula of the computational complexity matrix is as follows:

$$\begin{aligned} O(\cdot) &= \left[3n(hd)^2 + 2hn^2d + n(hd)^2\right] * L_B + 2n \times d_2\,(k+1) \\ &\quad + n^2 + 2n \times d_1 d_2 + n^2 + 4n \times d_2 + n^2 + n \times (d_2)^2 \\ &= n^2\left(2hdL_B + 3\right) \\ &\quad + n[4(hd)^2 L_B + 2d_2\,(k+1) + 2d_1 d_2 + 4d_2 + (d_2)^2] \end{aligned} \tag{10}$$

In addition, we conduct comparative analysis experiments on the number of parameters and computational time costs for the CDGAN-BERT model, as presented in Section 4.5.8.

The overall training and inference process algorithm based on our CDGAN-BERT is presented in Algorithm 1. The algorithm mainly consists of two parts: Training Stage, where lines 2 to 8 calculate the loss and perform iterative optimization of the model; Inference Stage, where lines 10 and 11 discard G and D^{rf} modules, then perform class probability prediction.

Algorithm 1 CDGAN-BERT Algorithm

Input: D_L: the Labeled Data; D_U: the Unlabeled Data;
Output: Model θ
1: **procedure** *Training Stage*:
2: Initialize Model θ
3: **while** not converged **do**
4: calculate the Adversarial Constraint Calculation loss $L_{G_{CoCa}}$ by Eq. (1)
5: calculate the Diversity Discriminator loss: $L_{D^{rf}_{unsupervised}}$ by Eq. (2) and L_D by Eq. (3)
6: calculate the G loss: L_G by Eq. (6)
7: update the Model $\theta \leftarrow L_D, L_{D^{rf}_{unsupervised}}, L_G$
8: **end while**
9: ***Return***: CDGAN-BERT θ.
10: **end procedure**
11: **procedure** *Inference Stage*:
12: discard G and D^{rf} from the trained CDGAN-BERT
13: ***Return***: Probabilistic Prediction of Classes
14: **end procedure**

4. Experiments

4.1. Datasets

To verify the performance of our proposed CDGAN-BERT, we conduct experiments on 6 publicly available text classification benchmark datasets. IMDB [34] dataset is widely used for binary sentiment classification of movie reviews. AG-News [35] is for the topic classification of news articles contains four categories. Yelp-5 [11] is a dataset containing five sentiment rating labels. Question Classification (QC) on the UIUC dataset [36] has two different grained setting: in the coarse-grained setting 6 classes are involved (QC-Coarse); in the fine-grained scenario the number of classes is 50 (QC-Fine). SST-2 [37] belongs to the text sentiment classification of a single sentence. More statistical information about datasets is summarized in Table 2. For all datasets except the QC-Coarse and QC-fine datasets, we randomly select the number of labeled data (N_{l-per}) to be 50, 100, 200, 500, 1000, and 2500, respectively.

To explore more deeply the performance of the model under extremely limited supervised. We adopt the same strategy as GAN-BERT to select annotated examples proportionally on QC-Coarse and QC-fine datasets. For instance, when only 1% labeled data ($N_{l-rate} = 0.01$) is used, i.e., every class contains one sample. For the selection of the number of unlabeled data, the ratio of unlabeled samples to labeled samples in each training set is $|N_U| = 100|N_L|$ (when available). Reference to GAN-BERT, to avoid divergences due to the unsupervised component of the adversarial training, we guarantee the presence of some labeled instances in each batch by replicating the labeled examples of a factor $log(|N_U|/|N_L|)$. In order to avoid the unfairness of data selection, for all datasets, we use the publicly random strategy[1] to repeat five times to select labeled samples as train set while keeping the original test set.

[1] We used the train/test split available within scikit-learn.

Table 2
Summary statistics for text classification datasets. The columns of Train, Unlabeled and Test are the number of samples; K is the number of categories.

Dataset	Train	Unlabeled	Test	K
IMDB	25,000	100,000	25,000	2
AG-News	120,000	100,000	7,600	4
Yelp-5	650,000	200,000	50,000	5
SST-2	67,349	20,000	872	2
QC-Coarse	5,400	–	500	6
QC-Fine	5,400	–	500	50

4.2. Implementation details

We implement in TensorFlow[2] by extending the SS-GAN framework. Specifically, D, D^{rf}, and G were all implemented as an MLP with one hidden layer activated by a $Leaky-Relu$ function, where $dropout = 0.1$. They all use the Adam optimizer [38] with a learning rate of $2e-5$. The pre-trained BERT adopts the same parameters as those provided on the official website.[3] Most of the Hyperparameter settings are: $Epochs = 6$, $\beta = 0.1$, $\gamma = 0.1$, $\tau = 0.3$, except AG-news ($N_{l-per} = 25$, Epochs = 6, $\beta = 0.08$, $\gamma = 0.13$, $\tau = 0.3$) and Yelp-5 ($N_{l-per} = 20$, Epochs = 6, $\beta = 0.1$, $\gamma = 0.13$, $\tau = 0.3$). On IMDB and Yelp-5 datasets, set the Maximum Sentence Length (MaxLength) = 300 and the Batchsize = 16. On other datasets, the MaxLength = 128 and the Batchsize = 32. All experiments are carried out on the Linux server with NVIDIA GeForce RTX 2080Ti GPU (12G), NVIDIA GeForce RTX 3090Ti GPU (24G), and NVIDIA Tesla V100 (32G). We repeated the experiment five times for all datasets containing the specified number of labeled samples and took the average as the final experimental result.

4.3. Baselines

We compare our CDGAN-BERT with previous state-of-the-art semi-supervised text classification approaches: VAMPIRE [5], UDA [11], VAT [1], S^2CL [39] and S2TC-BDD [9], using the performance reported by these papers. We also included the results from the SS-GAN-based baseline Methods: GAN-BERT [10] and SAT [40]. For BERT [41] and GAN-BERT [10], we conduct experiments with the same default parameters as GAN-BERT settings. In experiments, we utilize two different types of the averaged F1 score evaluation metrics, Micro-F1 and Macro-F1 (the public TF Metrics tool[4]).

The description of Baselines are as follows:

- VAMPIRE: A semi-supervised text classification method based on variational pre-training.
- BERT: A supervised text classification method built on the pre-trained BERT-based-uncased model1 and fine-tuned with the supervised softmax loss on labeled texts.
- UDA: A semi-supervised text classification method based on unsupervised data augmentation with back translation.
- GANBERT: A semi-supervised text classification method on Semi-Supervised Generative Adversarial Networks. In experiments, we utilize the default parameters, perform Generative Adversarial process with labeled samples.
- S2TC-BDD: A semi-supervised text classification method built on BERT with AM loss, namely Semi-Supervised Text Classification with Balanced Deep representation Distributions.
- SAT: An instance-adaptive self-training method for semi-supervised text classification. In experiments, we only use German as an intermediate language for data augmentation, and utilize the same parameter settings as the BERT and GANBERT baselines.

[2] https://github.com/tensorflow
[3] https://github.com/google-research/bert
[4] Multi-class metrics for Tensorflow, similar to scikit-learn multi-class metrics.

Table 3
Experimental results of the Micro-F1 scores on four text classification datasets. Where N_{l-per} means each class contains a different amount of labeled data, e.g., N_{l-per} : 50, 100, 200, 500, 1000, 2500. The best result are highlighted in boldface. And the ‡ denotes that the performance improvement of the CDGAN-BERT is statistically significant (paired samples t-test) at 0.01 level.

Dataset	AG-News						Yelp-5					
N_{l-per}	50	100	200	500	1000	2500	50	100	200	500	1000	2500
VAMPIRE	70.5$^{\dagger(25)}$	–	84.5$^{\dagger(250)}$	–	–	88.0†	22.7$^{\dagger(20)}$	–	47.6†	–	–	55.1$^{\dagger(2k)}$
UDA	85.5$^{\dagger(25)}$	–	88.3$^{\dagger(250)}$	–	–	90.6†	38.7$^{\dagger(20)}$	–	55.4†	–	–	58.0$^{\dagger(2k)}$
VAT	86.8$^{\dagger(25)}$	–	88.6$^{\dagger(250)}$	–	–	89.8†	24.4$^{\dagger(20)}$	–	55.1†	–	–	56.6$^{\dagger(2k)}$
BERT	83.948‡	85.327‡	86.447‡	87.539‡	89.829	91.355	43.491‡	47.994‡	52.753‡	54.742‡	56.203‡	56.658‡
GAN-BERT	85.658‡	85.908‡	87.384‡	88.250	89.579	91.750	45.089 ‡	48.552‡	52.028‡	55.816	56.674‡	58.175‡
S2TC-BDD	87.2$^{\dagger(25)}$	–	88.9$^{\dagger(250)}$	–	–	90.7†	41.7$^{\dagger(20)}$	–	55.2†	–	–	58.3$^{\dagger(2k)}$
S^{2}CL	86.6$^{\dagger(25)}$	–	**89.2**$^{\dagger(250)}$	–	–	–	43.8$^{\dagger(20)}$	–	55.4†	–	–	–
SAT	87.500	88.403	88.935	89.188	89.895	91.852	44.830‡	49.696‡	53.036‡	55.218‡	56.482‡	58.729‡
CDGAN-BERT(our)	**87.908**	**88.661**	89.026	**89.224**	**90.342**	**92.211**	**47.491**	**52.028**	**55.926**	**56.422**	**58.036**	**59.958**
Dataset	IMDB						SST-2					
VAMPIRE	–	82.2†	84.5$^{\dagger(250)}$	–	85.4$^{\dagger(1.25k)}$	87.1$^{\dagger(5k)}$	–	–	–	–	–	–
BERT	59.404‡	77.124‡	82.068‡	82.556‡	84.192‡	86.256‡	53.383‡	81.020‡	81.091‡	81.163‡	87.156‡	88.106‡
GAN-BERT	77.056‡	78.374‡	81.764‡	82.348‡	86.520‡	87.640‡	77.809‡	78.555‡	82.626‡	84.862‡	86.869‡	87.959‡
SAT	70.340‡	81.556‡	82.436‡	83.896‡	87.152‡	88.031‡	78.332‡	82.948‡	83.144‡	85.640‡	87.004‡	88.552‡
CDGAN-BERT(our)	**80.747**	**83.945**	**85.124**	**86.738**	**88.190**	**89.946**	**81.709**	**84.174**	**86.124**	**88.416**	**88.475**	**89.450**

Table 4
Experimental results of the Macro-F1 scores on four text classification datasets. Where N_{l-per} means each class contains a different amount of labeled data, e.g., N_{l-per} : 50, 100, 200, 500, 1000, 2500. The best result are highlighted in boldface. And the ‡ denotes that the performance improvement of the CDGAN-BERT is statistically significant (paired samples t-test) at 0.01 level.

Dataset	AG-News						Yelp-5					
N_{l-per}	50	100	200	500	1000	2500	50	100	200	500	1000	2500
VAMPIRE	69.8$^{\dagger(25)}$	–	83.3$^{\dagger(250)}$	–	–	87.6†	14.4$^{\dagger(20)}$	–	47.6†	–	–	55.3$^{\dagger(2k)}$
UDA	85.5$^{\dagger(25)}$	–	88.3$^{\dagger(250)}$	–	–	90.6†	35.7$^{\dagger(20)}$	–	**55.0**†	–	–	57.6$^{\dagger(2k)}$
VAT	86.7$^{\dagger(25)}$	–	88.6$^{\dagger(250)}$	–	–	89.7†	19.7$^{\dagger(20)}$	–	54.8†	–	–	56.9$^{\dagger(2k)}$
BERT	81.181‡	82.934‡	84.566‡	87.800‡	89.743	90.824‡	32.400‡	40.261‡	46.656‡	53.200‡	54.548‡	56.770‡
GAN-BERT	85.964‡	86.835‡	87.491‡	88.258‡	89.792	91.633	34.503‡	41.638‡	48.674‡	51.854‡	55.309	56.688‡
S2TC-BDD	87.2$^{\dagger(25)}$	–	88.9$^{\dagger(250)}$	–	–	90.7†	40.3$^{\dagger(20)}$	–	**55.0**†	–	–	**58.6**$^{\dagger(2k)}$
S^{2}CL	86.6$^{\dagger(25)}$	–	**89.2**$^{\dagger(250)}$	–	–	–	42.3$^{\dagger(20)}$	–	54.6†	–	–	–
SAT	87.454	88.136	89.084	89.247	89.768	91.639	38.643‡	40.984‡	50.152‡	52.254‡	54.780‡	55.647‡
CDGAN-BERT(our)	**87.897**	**88.548**	89.141	**89.573**	**90.555**	**92.090**	**43.854**	**46.331**	51.294	**53.590**	**56.004**	57.788
Dataset	IMDB						SST-2					
BERT	39.597‡	51.18‡	54.711‡	55.035‡	56.127‡	57.504‡	49.173‡	52.057‡	55.473‡	57.009‡	57.789‡	59.855
GAN-BERT	51.120‡	54.402‡	56.175‡	57.326	58.651	58.475‡	52.158‡	55.795	57.075	57.686	58.123‡	59.511
SAT	54.795	56.547	57.435	**58.381**	59.194	59.982	52.038‡	54.925‡	56.137‡	57.635‡	**59.493**	**60.124**
CDGAN-BERT(our)	**55.632**	**57.011**	**57.893**	58.229	**59.214**	**60.170**	**55.970**	**56.425**	**57.577**	**58.268**	59.203	59.705

• S^{2}CL: A semi-supervised text classification method via Self-paced Semantic-level Contrastive Learning, which use semantic-level contrastive learning modules and robust supervised learning to improve the quality of pseudo-labels.

4.4. Main results

Before introducing experiment results, the data with † in the upper right corner represents which reported by these papers. The numerical value to the right of † indicates the number of labeled samples. The best scores are high-lighted in boldface.

4.4.1. Varying number of labeled texts

First, we evaluate the performance of the CDGAN-BERT on data containing different numbers of labels. On the four data of AG-News, IMDB, Yelp-5, and SST-2, the experimental results of Micro-F1 and Macro-F1 are shown in Tables 3 and 4, respectively. Generally speaking, our proposed CDGAN-BERT outperforms the baselines and reaches a state-of-the-art result in most cases.

From the experimental results, we can observe that our method improves the most in terms of Micro-F1 and Macro-F1 scores compared to VAMPIRE and pre-trained BERT methods, with an average improvement of 6.899%, 9.438% and 4.754%, 5.661%, respectively. Especially when the labeled texts are scarce, our approach achieves a big margin. For example, when label = 50, on the AG-News and Yelp-5 datasets, Micro-F1 increased by 16.816%, 23.791%, and Macro-F1 increased by 17.554% and 16.465%, respectively.

Compared to the S^{2}CL method, our proposed method improved by 1.334% on average of Micro-F1 scores. However, for the AG-News dataset (N_{l-per} = 200), we observed a decrease of 0.174% in the Micro-F1 score and 0.059% in the Macro-F1 score, and 3.306% lower on the Yelp dataset (N_{l-per} = 200). In all cases except for the ones mentioned above, CDGAN-BERT demonstrate better classification performance compared to the S^{2}CL method. The possible reasons for the observed degraded classification performance are twofold. Firstly, the relatively small amount of labeled data may have resulted in a slight reduction in classification scores. Secondly, our model might lack the ability to accurately recognize specific classes in certain datasets without the use of data augmentation techniques.

Compared to the SAT baseline, our proposed method achieves average improvements of 2.343% and 1.145% in the Micro-F1 and Macro-F1 scores, respectively. The most significant improvement is observed on the Yelp-5 dataset, where the scores improve by 3.478% and 2.733%, respectively. And the improvement on the AG-News dataset is relatively modest, with scores increasing by only 0.413% and 0.227%, respectively. However, on the SST-2 dataset, when the number of labeled texts was set to 1000 and 2500, our proposed method was 0.29% and 0.419% lower than the SAT baseline on the Macro-F1 scores, respectively. We think that the reason for this discrepancy is that on the SST-2 dataset, the SAT baseline utilizes data augmentation to better improve the classification performance of the model when the number

Table 5
Experimental results of Micro-F1 and Macro-F1 scores under extremely limited supervision, where each class contains different ratios of labeled data, e.g., N_{l-rate} : 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5. The best result are highlighted in boldface. On two QC tasks: the fine-grained setting of 50 categories and the coarse-grained setting of 6 categories. QC-Fine: 50 categories. QC-Coarse: 6 categories.

Dataset	QC-Fine								QC-Coarse							
N_{l-rate}	0.01	0.02	0.05	0.1	0.2	0.3	0.4	0.5	0.01	0.02	0.05	0.1	0.2	0.3	0.4	0.5
Micro-F1 Scores																
BERT	1.80	13.6†	25.4	37.0	56.8	60.0	65.7	68.8	0.442	65.4	81.4	92.8	94.2	95.0	95.4	96.2
GAN-BERT	34.0	41.8†	63.6	68.1	73.4	75.2	76.6	77.4	58.3	80.4	90.7	94.0	95.9	96.2	96.4	96.4
CDGAN-BERT(our)	**52.6**	**57.8**	**69.3**	**74.4**	**79.8**	**80.6**	**81.8**	**83.7**	**73.0**	**87.3**	**92.9**	**95.8**	**96.3**	**96.9**	**96.9**	**97.2**
Macro-F1 Scores																
BERT	0.141	1.04	1.076	2.796	9.978	13.492	25.003	23.533	5.267	15.993	26.375	50.909	72.199	73.976	76.922	80.732
GAN-BERT	7.597	12.56	21.264	25.612	34.061	32.176	34.202	37.030	44.450	64.154	72.213	76.440	80.764	79.959	81.034	81.146
CDGAN-BERT(our)	**9.526**	**14.240**	**27.223**	**32.951**	**36.974**	**38.627**	**41.427**	**42.122**	**55.733**	**64.252**	**77.751**	**81.346**	**82.512**	**82.695**	**83.090**	**83.399**

of labeled texts is large. Nonetheless, in all other cases, CDGAN-BERT achieves higher classification scores. The above results demonstrate that our approach can achieve comparable or even better performance in semi-supervised text classification compared to the instance-adaptive self-training method.

Compared with UDA and S2TC-BDD methods, our method still achieves good improvement in most cases, except when the AG-News dataset $N_{l-per} = 200$, the result in the Macro-F1 metric is worse than the S2TC-BDD method. We think the reason may be that the model is highly recognizable for specific categories and lower for other categories, resulting in a lower Macro-F1 score. But we can also get excellent results compared to BERT and GANBERT methods, which show our model remains very competitive. Moreover, for fairness, we supplement some experiments. On AG-news, when $N_{l-per} = 25$, the Micro-F1 and Macro-F1 scores are **87.487%** and **87.416%**, respectively. And, on the Yelp-5 dataset, when $N_{l-per} = 20$ and $N_{l-per} = 2000$, the Micro-F1 scores are **42.666%** and **59.050%**, respectively. These results show that our model still outperforms the baseline under the same experimental setting. In summary, the experiments show that our model with adversarial constraint and a diversity discriminator achieves excellent performance in the semi-supervised text classification.

Summarizing the above experimental results and analysis, our proposed CDGAN-BERT method achieves more higher Micro-F1 and Macro-F1 classification scores in the vast majority of cases compared to several different baseline models. In addition, the paired samples t-test statistical significance results show that on Micro-F1 scores, CDGAN-BERT shows better significance on Yelp-5, IMDB and SST-2 datasets and has better and more stable classification performance. On the AG-News dataset, the significance performance is less favorable. In terms of Macro-F1 scores, CDGAN-BERT likewise shows better significance on the Yelp-5 and SST-2 datasets. On the AG-News and IMDB datasets, significance performs less well. We believe that the low Macro-F1 scores are caused by the relatively poor discrimination ability of the CDGAN-BERT model on some specific categories. Compared to the SAT baseline, the CDGAN-BERT model achieved high categorization scores on most of the datasets, although it performed poorly in terms of significance. Besides, we believe that the main reason for the relatively poor classification performance on the AG-News dataset may be the shorter average sentence length and higher number of categories in this dataset, which makes it difficult for the model to learn the classification information. However, compared with the semi-supervised text categorization by means of data augmentation, our method does not rely on other additional trained models and achieves the same competitive or higher classification scores with this class of methods. Therefore, we believe that the CDGAN-BERT method can effectively improve the model's classification performance on semi-supervised text classification tasks.

4.4.2. Learning with a scarce amount of labeled data

Based on the previous verification, we further explored the model's performance in the case of a very scarcity of labeled data. So, we further conduct experiments on the QC-Coarse and QC-Fine datasets divided by proportion. The experimental results are shown in Table 5. On the QC task, whether it is a coarse-grained classification or a more subtle fine-grained classification. Compared to BERT and GAN-BERT baselines, our model has the best classification performance in all cases. Especially when labeled samples are extremely scarce, the performance improvement is more significant. In the case of minimal labeled data, to avoid divergence due to the unsupervised part of adversarial training, it is guaranteed that there are some labeled examples in each batch. We use the same settings as GAN-BERT, in that work, they replicated the labeled examples of a factor $log(N_u/N_l)$. In detail, compared with the BERT baseline, Micro-F1 and Macro-F1 have an average improvement of 20.16% and 23.45%, respectively. Compared with the GAN-BERT baseline, Micro-F1 and Macro-F1 have an average increase of 6.75% and 3.83%, respectively. The improvement is noticeable when $N_{l-rate} = 0.01$, the Micro-F1 scores improve by 50% and 18% over the BERT and GAN-BERT baselines.

4.5. Ablation study

In this section, we conduct extensive studies from different perspectives to better understand our CDGAN-BERT method. More studies are presented in detail in the next subsections.

4.5.1. Effect of D^{rf} and $L_{G_{CoCa}}$

To compare the influence of discriminator D^{rf} or the model's intrinsic adversarial constraints $L_{G_{CoCa}}$. We perform ablation studies on the QC-fine and QC-coarse datasets and analyze the contributions of the different component terms present in the CDGAN-BERT. The experimental results are shown in Table 6. In all situations, adding the discriminator D^{rf} or putting the adversarial constraint on the model's intrinsic state representation $L_{G_{CoCa}}$ could significantly improve the model's performance. In most cases, the two together will achieve the best results. Except for a few cases, such as on the QC-Coarse task, adding $L_{G_{CoCa}}$ achieves the optimal outcome when the labeled ratio is 0.4 and 0.5.

4.5.2. Labeled rate and the model's performance

To observe the possible impact relationship between the amount of labeled data and model performance, we conduct comparative experiments with BERT and GAN-BERT baselines. Figs. 2 and 3 show the model performance varies with the Labeled rate on QC-Fine and QC-Coarse Text Classification. Compared with BERT and GAN-BERT, on the QC-Fine and QC-coarse datasets, our proposed method improves 50.8%, 18.6% and 28.8%, 14.7% when the label ratio is 0.01, respectively, and 14.9%, 6.3% and 1.0%, 0.8%, when the label ratio is 0.5, respectively. When the number of labeled data is small, the CDGAN-BERT model shows a more significant improvement in classification scores with the addition of even a small amount of labeled data. Conversely, when the number of labeled data is already large, the improvement in classification scores becomes less pronounced when

Table 6
Ablation studies: Classification performance (Micro-F1 score (%)) on QC-fine and QC-Coarse datasets with different ratios of labeled data after removing different modules of CDGAN-BERT.

Dataset	QC-Fine								QC-Coarse							
N_{l-rate}	0.01	0.02	0.05	0.1	0.2	0.3	0.4	0.5	0.01	0.02	0.05	0.1	0.2	0.3	0.4	0.5
CDGAN-BERT	**52.6**	**57.8**	**69.3**	**74.4**	**79.8**	**80.6**	**81.8**	**83.7**	**73.0**	**87.3**	**92.9**	**95.8**	**96.3**	**96.9**	96.9	97.2
w/o D^{rf}	50.4	57.0	68.3	72.8	77.7	78.5	79.7	80.2	75.4	86.6	89.9	94.3	95.9	96.8	***97.1***	***97.3***
w/o $L_{G_{CoCa}}$	49.0	52.2	65.4	71.2	76.4	77.0	78.2	79.6	70.8	85.2	92.6	92.4	95.8	96.0	96.0	96.4
w/o $L_{G_{CoCa}}$&D^{rf}	34.0	41.8	63.6	68.1	73.4	75.2	76.6	77.4	58.3	80.4	90.7	94.0	95.9	96.2	96.4	96.4

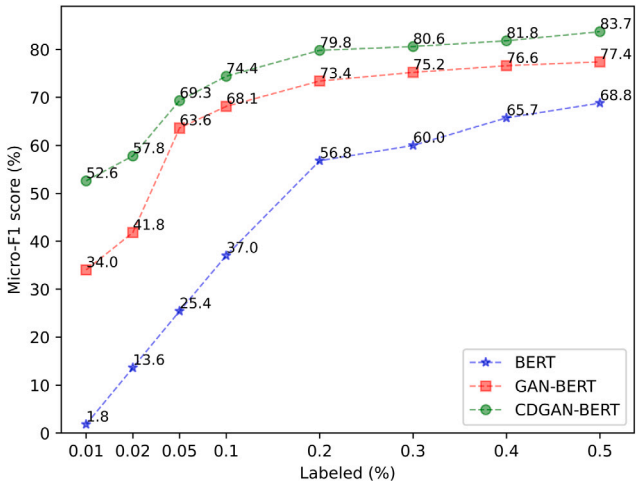

Fig. 2. QC-Fine Granied.

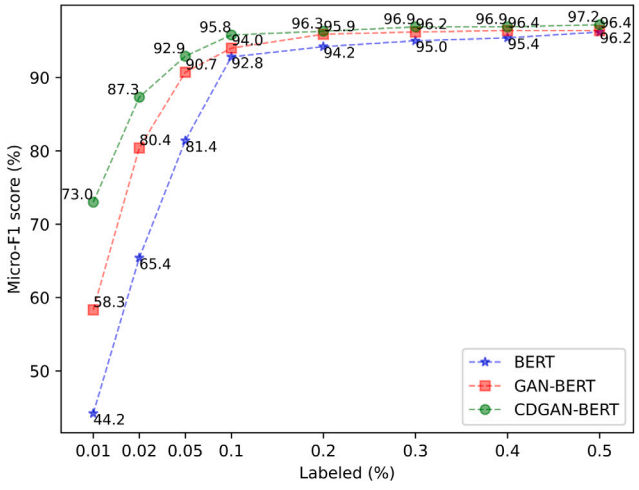

Fig. 3. QC-Coarse Grained.

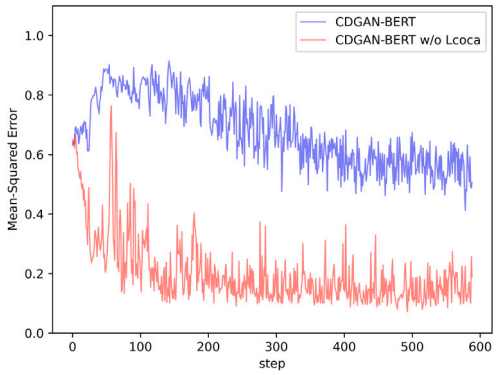

Fig. 4. MSE.

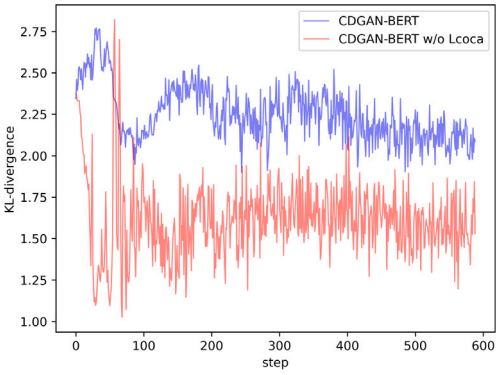

Fig. 5. KL divergence.

additional labeled data is added. For example, on the QC-Fine dataset, when the labeled ratio was increased from 0.01 to 0.02, the number of each class containing labeled data was increased from 1 to 2, the Micro-F1 score was improved by 5.2%. The Micro-F1 score increased by only 1.2% when the percentage of labeled data increased from 0.3 to 0.4. The above results demonstrate the effectiveness of CDGAN-BERT on semi-supervised text classification, especially when the amount of labeled data is extremely limited. When the number of labeled data is small, CDGAN-BERT shows significant improvements in classification performance. However, as the amount of labeled data increases, the improvement in model classification performance becomes less prominent.

4.5.3. Distribution difference between P_G and P_B

On the QC-Fine dataset, we measure the distributional differences of P_G and P_B produced every ten steps during the training process by using Mean Squared Error (MSE) and Kullback–Leibler divergence (KL divergence), respectively. As shown in Figs. 4 and 5, compared with removing $L_{G_{CoCa}}$, the distribution difference between P_G and P_B is slightly bigger, but the overall difference shows a downward trend and gradually tends to a generally stable state; simultaneously, the amplitude of the oscillation decrease. This result indicates that the P_G generated by the generator will continue to produce more stable adversarial effects after adding adversarial constraints. And this 'adversarial' will facilitate the model to learn more feature information.

4.5.4. Effect of hyperparameter τ

Fig. 6 shows the effect of Hyperparameter $\tau \in (0.01, 1)$ on QC-Fine Grained dataset. We conduct experiments on two labeled rate 0.02 and 0.2, respectively. The Micro-F1 scores of CDGAN-BERT are plotted by the blue asterisks and red grids in Fig. 6. In addition, the gray and yellow dashed lines represent the Micro-F1 scores on the GAN-BERT baseline. Experiments show that the adversarial constraint on the model's internal representation has significantly improved the model's classification performance. And the adversarial constraint has

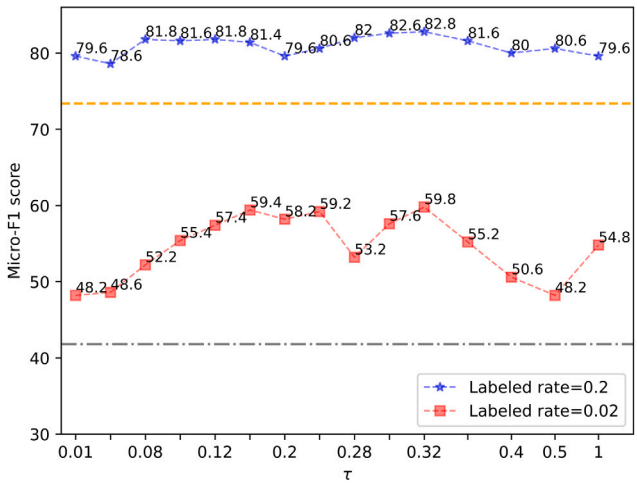

Fig. 6. Hyperparameter τ.

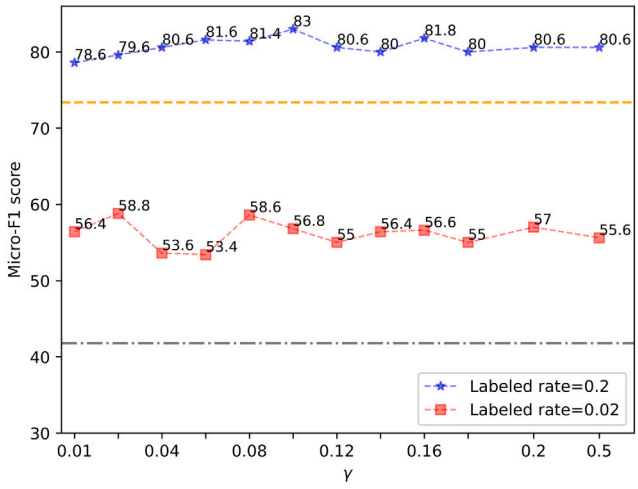

Fig. 7. Hyperparameter γ.

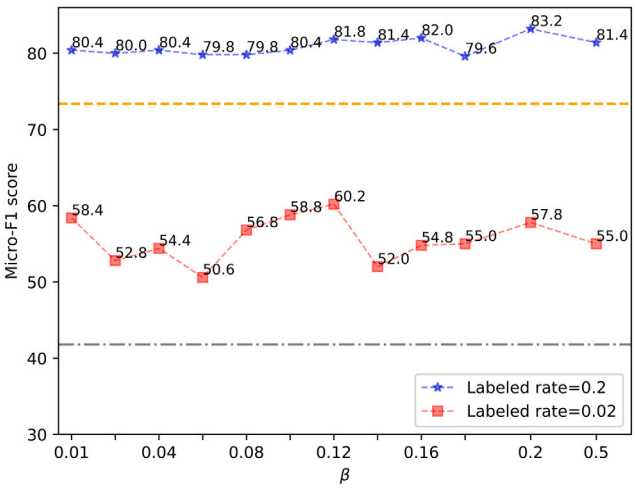

Fig. 8. Hyperparameter β.

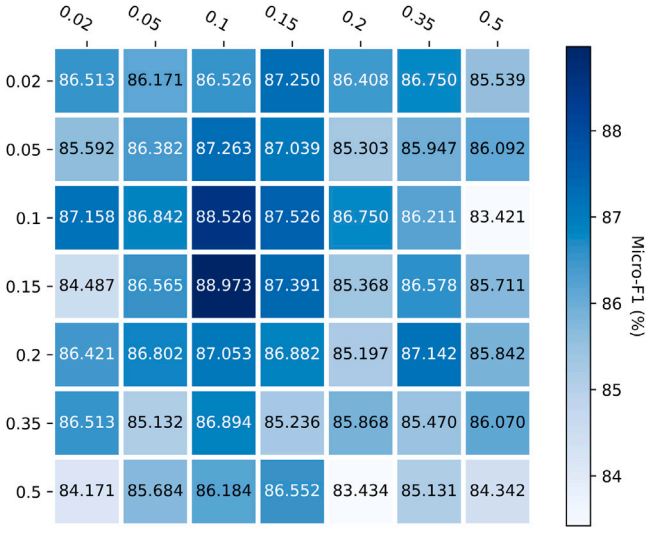

Fig. 9. Classification performance on *AG-News* dataset with 50 labeled samples under different hyperparameters γ and β.

a more profound impact when the number of labeled samples is small than when the number of labeled samples is large. We think the model already has good performance with a large number of labeled samples. Although the adversarial constraints can furtherly improve model performance, the improvement is limited.

4.5.5. Effect of hyperparameter γ or β

Figs. 7 and 8 show the effect of Hyperparameter $\gamma \in (0.01, 0.5)$ and $\beta \in (0.01, 0.5)$ on QC-Fine Grained dataset. The illustration is the same as that in Fig. 6. Fig. 7 shows that D^{rf} acting on G will also improve the model's performance. We believe that the reason is that when D^{rf} acts on the generator, it will prompt G to generate more realistic samples, further improving the overall performance of the model. And Fig. 8 shows that D^{rf} acting on D will also improve the model's performance. We think the reason is that D^{rf} and D will refine the tasks with different attributes after adding D^{rf}, which will assist the discriminator D in improving the classification ability.

To better evaluate the effect of hyperparameters γ and β, we conducted ablation experiments on the *AG-News* (N_{l-per} : 50) and *QC-Fine* (N_{l-rate} : 0.2) datasets. we choose two hyperparameters from the set of values [0.02, 0.05, 0.1, 0.15, 0.2, 0.35, 0.5] for detailed analysis. The results of these experiments are presented in heatmaps, as depicted in Figs. 9 and 10. In the heatmaps, the horizontal axis represents parameter γ, while the vertical axis represents parameter β. As depicted in the figure, darker colors correspond to better classification performance, and the results show that values around 0.1 for both parameters produce better results. Specifically, the highest classification scores are achieved when γ and β are set to 0.1 and 0.15 on both datasets. Deviating too much from these values, either towards larger or smaller values, leads to suboptimal classification performance. These results provide evidence that the auxiliary discriminator has a positive impact on both the generator and the discriminator, ultimately enhancing the overall classification performance of the model.

4.5.6. The selection of hyperparameters β and γ

In addition to conducting experiments on the impact of hyperparameters on QC-Fine Grained dataset, we also verify the selection of hyperparameters β and γ on dthe AG-News dataset. The Micro-F1 scores on the AG-News dataset (N_{l-per} = 50) as shown in Table 7. Experimental results show that when hyperparameters β and γ take non-zero values, the Micro-F1 score is higher than when they take zero. At the same time, when the value is 0.1, the score is better, so we set the hyperparameters β and γ to 0.1 in all dataset experiments.

Table 7
The Micro-F1 scores on the AG-News dataset (N_{l-per} = 50).

#Hyperp	0	0.01	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.2	0.5	1
β	85.658	88.053	87.868	86.934	87.5	85.816	87.84	87.697	86.158	88.316	87.329	87.711	87.302	87.408
γ	85.658	86.263	86.566	87.776	88.145	87.671	87.487	88.211	87.921	87.724	87.605	88.316	88.105	87.908

Table 8
Comparison of the efficiency.

Model	#Params	# Computing time costs
BERT	325.05M	2 min
GAN-BERT	328.82M	10 min
CDGAN-BERT	331.78M	11 min

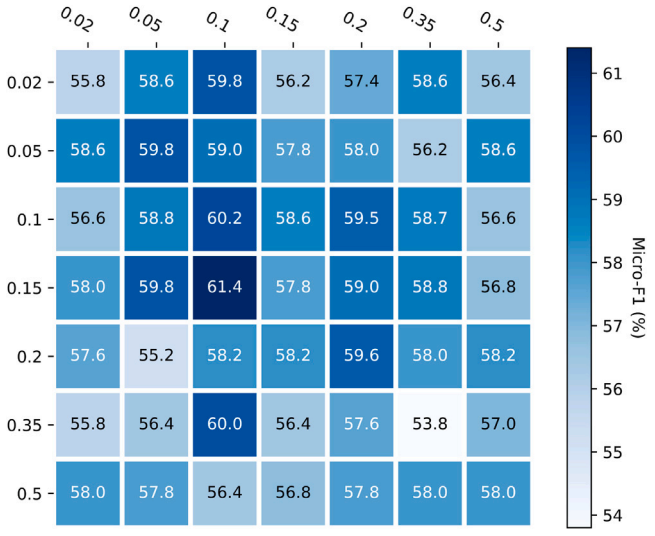

Fig. 10. Classification performance on *QC-Fine* dataset with a labeled ratio of 0.2 under different hyperparameters γ and β.

4.5.7. The effect of the epochs

We conducted experiments on the *QC-Fine* dataset with two different label ratios (0.02 and 0.2). As shown in Fig. 11, the Micro-F1 score gradually increased with the increase of Epoch, and basically reached a stable level after the sixth Epoch. On both datasets, when the Epoch is less than 6, the classification score increases significantly, and the model is obviously not optimal. Specifically, on *QC-Fine* with a labeled ratio of 0.02, the model achieves the highest Micro-F1 score when the Epoch is set to 10. Conversely, on *QC-Fine* with a labeled ratio of 0.2, the highest Micro-F1 scores are obtained when the Epoch is either 6 or 8. It can be seen from the experimental results that although our method will fluctuate a little with the Epoch increasing, it will tend to be stable as a whole. In this work, to avoid excessive computational time, most of the Hyperparameter Epoch settings are 6.

4.5.8. Comparison of the efficiency

We experiment on the AG-News dataset (N_{l-per} = 50, Epochs = 3) with NVIDIA Tesla V100 (32G), and the complex issues and computational costs are shown in the Table 8. Compared with BERT and GAN-BERT, the complexity of CDGAN-BERT is almost the same. Regarding the computation costs, our model computes more time cost than BERT, but nearly the same as GAN-BERT.

5. Conclusion

In this paper, we propose a novel CDGAN-BERT, a model includes the adversarial constraint on the model's intrinsic state representation and the diversity discriminator. Unlike other works, we were more concerned the samples generated by the generator should approximate

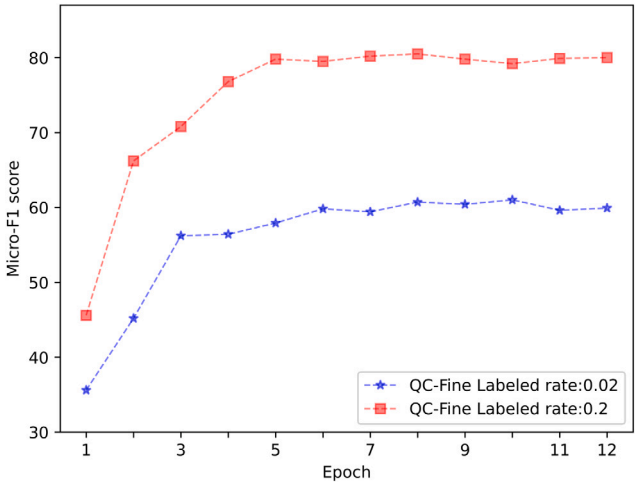

Fig. 11. Effect of Epochs.

the real examples and remain adversarial. To our knowledge, we are the first to introduce adversarial constraints on the model's intrinsic state representation using the reciprocal of MSE. The purpose of this constraint to help the model learn higher-quality data representations from 'adversarial'. Then, we design a diversity discriminator to improve the model's identification ability and classification performance by mitigating discriminative ambiguity. In this work, we validate our model on 6 text classification datasets and achieve excellent performance. Overall, we believe that such SS-GAN-like algorithms which introduce the adversarial constraint and the diversity discriminator will facilitate the application of machine learning in more and more practical domains where labels are expensive or difficult to obtain. In this paper, we only demonstrate the effectiveness of the CDGAN-BERT method on a semi-supervised text classification task. However, we have not yet applied it to other tasks such as translation, entity recognition, and relationship extraction. In future work, we will apply our proposed method to other NLP tasks and even extend it to the image processing domain.

Broader Impact: In previous work about GAN and SS-GAN, researchers paid more attention to the more realistic samples generated by the generator. Our work improves the model's classification performance by enhancing the BERT's representation ability. The generated data could help BERT learn more high-quality representation. We propose to directly impose adversarial constraints on the distribution of generated samples to make the generated samples have 'adversarial' (richer diversity) while maintaining reality. These 'adversarial' examples can promote BERT to achieve higher quality representations. CDGAN-BERT does not rely on additional models for data augmentation (such as back-translation, and interpolation-based data augmentation). We believe that CDGAN-BERT provides a new solution for the classification task with only a few annotated samples during model training.

CRediT authorship contribution statement

Nai Zhou: Writing – original draft, Visualization, Validation, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Nianmin Yao:** Writing – review & editing,

Supervision, Methodology, Conceptualization. **Nannan Hu:** Writing – review & editing. **Jian Zhao:** Funding acquisition. **Yanan Zhang:** Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Innovation Foundation of Science and Technology of Dalian under Grant No. 2018J12GX045 and National Key R&D Program of China under Grant No. 2018AAA0100300 and Project of China National Intellectual Property Administration No. 220134.

References

[1] T. Miyato, S.-i. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2018) 1979–1993.
[2] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C.A. Raffel, E.D. Cubuk, A. Kurakin, C.-L. Li, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, Adv. Neural Inf. Process. Syst. 33 (2020) 596–608.
[3] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C.A. Raffel, Mixmatch: A holistic approach to semi-supervised learning, Adv. Neural Inf. Process. Syst. 32 (2019).
[4] T. Miyato, A.M. Dai, I. Goodfellow, Adversarial training methods for semi-supervised text classification, in: International Conference on Learning Representations, 2017.
[5] S. Gururangan, T. Dang, D. Card, N.A. Smith, Variational pretraining for semi-supervised text classification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5880–5894.
[6] D.S. Sachan, M. Zaheer, R. Salakhutdinov, Revisiting lstm networks for semi-supervised text classification via mixed objective function, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 01, 2019, pp. 6940–6948.
[7] C. Liu, Z. Mengchao, F. Zhibing, P. Hou, Y. Li, FLiText: a faster and lighter semi-supervised text classification with convolution networks, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 2481–2491.
[8] J. Chen, Z. Yang, D. Yang, MixText: linguistically-informed interpolation of hidden space for semi-supervised text classification, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2147–2157.
[9] C. Li, X. Li, J. Ouyang, Semi-supervised text classification with balanced deep representation distributions, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 5044–5053.
[10] D. Croce, G. Castellucci, R. Basili, 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, 2020, pp. 2114–2119.
[11] Q. Xie, Z. Dai, E. Hovy, T. Luong, Q. Le, Unsupervised data augmentation for consistency training, Adv. Neural Inf. Process. Syst. 33 (2020) 6256–6268.
[12] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, Adv. Neural Inf. Process. Syst. 29 (2016).
[13] J. Sun, B. Bhattarai, T.-K. Kim, MatchGAN: a self-supervised semi-supervised conditional generative adversarial network, in: Proceedings of the Asian Conference on Computer Vision, 2020.
[14] A. Haque, EC-GAN: low-sample classification using semi-supervised algorithms and GANs (student abstract), in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 18, 2021, pp. 15797–15798.
[15] S. Motamed, F. Khalvati, Multi-class generative adversarial nets for semi-supervised image classification, 2021, arXiv preprint arXiv:2102.06944.
[16] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Adv. Neural Inf. Process. Syst. 3 (2014) 2672–2680.
[17] C. Li, T. Xu, J. Zhu, B. Zhang, Triple generative adversarial nets, Adv. Neural Inf. Process. Syst. 30 (2017).
[18] J. Donahue, P. Krähenbühl, T. Darrell, Adversarial feature learning, 2016, arXiv preprint arXiv:1605.09782.
[19] S. Wang, J. Cai, Q. Lin, W. Guo, An overview of unsupervised deep feature representation for text categorization, IEEE Trans. Comput. Soc. Syst. 6 (3) (2019) 504–517.
[20] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P.S. Yu, L. He, A survey on text classification: From traditional to deep learning, ACM Trans. Intell. Syst. Technol. 13 (2) (2022) 1–41.
[21] M. Chen, Q. Tang, K. Livescu, K. Gimpel, Variational sequential labelers for semi-supervised learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 215–226.
[22] Z. Yang, Z. Hu, R. Salakhutdinov, T. Berg-Kirkpatrick, Improved variational autoencoders for text modeling using dilated convolutions, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 3881–3890.
[23] W. Guo, J. Cai, S. Wang, Unsupervised discriminative feature representation via adversarial auto-encoder, Appl. Intell. 50 (2020) 1155–1171.
[24] X. Dong, Y. Zhu, Y. Zhang, Z. Fu, D. Xu, S. Yang, G. De Melo, Leveraging adversarial training in self-learning for cross-lingual text classification, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1541–1544.
[25] R. Sennrich, B. Haddow, A. Birch, Improving neural machine translation models with monolingual data, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 86–96.
[26] J. Chen, Y. Wu, D. Yang, Semi-supervised models via data augmentation for classifying interactive affective responses, in: AffCon@ AAAI, 2020.
[27] Z. Dai, Z. Yang, F. Yang, W.W. Cohen, R.R. Salakhutdinov, Good semi-supervised learning that requires a bad gan, Adv. Neural Inf. Process. Syst. 30 (2017).
[28] J.T. Springenberg, Unsupervised and semi-supervised learning with categorical generative adversarial networks, 2015, arXiv preprint arXiv:1511.06390.
[29] A. Odena, Semi-supervised learning with generative adversarial networks, 2016, arXiv preprint arXiv:1606.01583.
[30] N. Zhou, N. Yao, J. Zhao, Y. Zhang, Rule-based adversarial sample generation for text classification, Neural Comput. Appl. (2022) 1–12.
[31] M. Gong, Y. Xu, C. Li, K. Zhang, K. Batmanghelich, Twin auxilary classifiers GAN, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.
[32] H. Zhang, Z. Zhang, A. Odena, H. Lee, Consistency regularization for generative adversarial networks, in: International Conference on Learning Representations, 2019.
[33] D. Croce, G. Castellucci, R. Basili, Kernel-based generative adversarial networks for weakly supervised learning, in: International Conference of the Italian Association for Artificial Intelligence, Springer, 2019, pp. 336–347.
[34] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, C. Potts, Learning word vectors for sentiment analysis, in: Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011.
[35] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, Adv. Neural Inf. Process. Syst. 28 (2015).
[36] X. Li, D. Roth, Learning question classifiers: the role of semantic information, Nat. Lang. Eng. 12 (3) (2006) 229–249.
[37] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, GLUE: a multi-task benchmark and analysis platform for natural language understanding, in: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, 2018, pp. 353–355.
[38] D. Kingma, J. Ba, Adam: a method for stochastic optimization, Comput. Sci. (2014).
[39] Y. Xia, K. Zhang, K. Zhou, R. Wang, X. Hu, Semi-supervised text classification via self-paced semantic-level contrast, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2023, pp. 482–494.
[40] H. Chen, W. Han, S. Poria, SAT: improving semi-supervised text classification with simple instance-adaptive self-training, in: Findings of the Association for Computational Linguistics: EMNLP 2022, 2022, pp. 6141–6146.
[41] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT (1), 2019.