Towards Better Detection of Biased Language with Scarce, Noisy, and Biased Annotations

Zhuoyan Li*
Purdue University
West Lafayette, Indiana, USA
li4187@purdue.edu

Zhuoran Lu*
Purdue University
West Lafayette, Indiana, USA
lu800@purdue.edu

Ming Yin
Purdue University
West Lafayette, Indiana, USA
mingyin@purdue.edu

ABSTRACT

Biased language is prevalent in today's online social media. To reduce the amount of online biased language, one critical first step is to accurately detect such biased language, ideally automatically. This is a challenging problem, however, as the annotated data necessary for training a biased language classifier is either scarce and costly (e.g., when collected from experts), or noisy and potentially biased on their own (e.g., when collected from crowd workers). The biased language classifier built based on these annotations may thus be inaccurate, and sometimes unfair (e.g., have systematic accuracy disparities across texts with different political leanings). In this paper, we propose a novel method, CLEARE, for biased language detection, in which we utilize self-supervised contrastive learning to enhance the biased language classifier—we learn a robust encoder of the textual data through solving a min-max optimization problem, so that the encoder could help achieve the best classification performance even if the worst data augmentation strategy is selected. Extensive evaluations suggest that CLEARE shows substantial improvements compared to the state-of-art biased language detection methods on several benchmark datasets, in terms of improving both the accuracy and the fairness of the detection.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Learning latent representations**.

KEYWORDS

Biased Language, Bias Detection, Contrastive Learning, Fairness

ACM Reference Format:
Zhuoyan Li, Zhuoran Lu, and Ming Yin. 2022. Towards Better Detection of Biased Language with Scarce, Noisy, and Biased Annotations . In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society (AIES'22), August 1–3, 2022, Oxford, United Kingdom.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3514094.3534142

1 INTRODUCTION

Today, biased language (e.g., language that is offensive, prejudiced, or hurtful) is increasingly widespread in the online social media, which makes the rapid detection and mitigation of online biased language a urgent need. However, due to the rapidly increasing magnitude of text resources on the Internet, it is impossible for human content moderators to traverse all the possible biased text manually [60]. In light of this, researchers have started to develop machine learning solutions, for instance, language models, to automatically detect the biased texts [46, 68].

However, obtaining a high-performing biased language detector is still quite challenging today. This is because the differences between biased texts and non-biased texts is often quite subtle [47, 48]. Thus, in order to train a model to automatically classify biased language, the quality requirement for annotations is high. To meet this requirement, a common approach adopted in previous studies [47] is to hire linguistic experts to offer their professional judgement, which usually will result in a set of high-quality annotated dataset of small scale, and it is usually exceedingly costly. An alternative approach to get annotations at a large scale is to outsource them to crowd workers (i.e., crowdsourcing). However, crowd workers' annotations are usually noisy, and sometimes reflect their own biases in interpreting the text. For instance, it is showed that people's partisanship significantly shapes their views of texts on social media, resulting people to believe texts aligning with their own political stance as non-biased and texts not aligning with their stance as biased [24]. Building a biased language detector based on these noisy and biased annotations may not only lead to an inaccurate detector, but perhaps an unfair detector that have systematic performance discrepancies across texts of different subgroups, such as text reflecting different political leanings [64, 66].

To deal with the issues of data scarcity, data noise, and data biases in biased language detection, a natural idea is to utilize self-supervised contrastive learning to enhance the biased language classifiers. Specifically, self-supervised contrastive learning can be used to learn the latent representation of the text data through data augmentations—which directly addresses the data scarcity concerns—and it also has the promise of capturing the subtle semantic information embedded in the text data that is beyond the signals provided by the direct supervision (i.e., the labels). However, applying self-supervised contrastive learning in the context of biased language detection is not as straight-forward as it seems, since if the data augmentation strategies are not chosen appropriately, the classifier may get misled by pairs of original and augmented texts that actually would have been associated with different labels.

Thus, in this paper, we propose a novel approach, Contrastive Learning with Robust Encoder (CLEARE), for biased language detection. In CLEARE, we aim to learn a *robust* encoder of the textual data via self-supervised contrastive learning, and the encoder is robust in the sense that it will ensure the best classification performance even if the data augmentation strategy is adversarially

*Li and Lu have made equal contributions to this work.

AIES'22, August 1–3, 2022, Oxford, United Kingdom

ACM ISBN 978-1-4503-9247-1/22/08.
https://doi.org/10.1145/3514094.3534142

selected to confuse the encoder. We turn the problem of learning this robust encoder into a min-max optimization problem. We then solve this problem by alternating between updating the encoder and supervised learning model parameters given a fixed data augmentation policy, and updating the data augmentation policy given a fixed set of model parameters.

We examine the effectiveness of CLEARE via a series of evaluation studies. In the first set of evaluations, on three banchmark datasets, we find that CLEARE consistently outperforms a wide range of baseline approaches in making accurate classification of biased language. Through two sets of simulation studies, we also find that the advantages of CLEARE over other baseline approaches are robust when we vary the size of the training dataset or as we change the levels of noise in the labels. Moreover, in a second set of evaluations, we conduct both simulation and real-world studies (with annotation data collected from crowd workers from Amazon Mechanical Turk) to explore the performance of CLEARE in advancing fair detection of biased language. Our results suggest that compared to the baseline approaches, CLEARE increases the equality of classifier performance on different subsets of texts which reflect different political leanings. In other words, CLEARE leads to fairer biased language detectors.

Our contributions can be summarized as follows:

- *Methodological:* We developed a novel contrastive learning framework for biased language detection. It has the potential to address the issues of data scarcity, data noise, and data biases, which are prevalent in biased language detection.
- *Experimental:* We conducted extensive experiments to demonstrate that the proposed method substantially outperforms the state-of-the-art methods on several benchmark datasets with respect to multiple evaluation metrics (e.g., accuracy, fairness).
- *Dataset:* We constructed a biased language annotation dataset, which was annotated by crowd workers with different political leanings[1]; this dataset can serve as a valuable benchmark for researchers to study annotation bias and evaluate the biased language detector's fairness levels across texts of different political leanings in the future.

2 RELATED WORK

Biased Text Mitigation. Biased text mitigation is an important problem in the NLP domain, which involves biased text classification [1, 18] and debiased text generation [45, 64, 67]. For biased text classification, traditional methods utilize handcrafted features or linguistic and lexical rules [3, 18, 27, 43, 57]. With the rise of deep learning, many deep neural models have been developed for classifying biased texts in an end-to-end manner [15, 37, 39, 46, 68]. For debiased text generation, some works [6, 32, 65, 66] consider it as the style transfer task [14, 17], which aim to reduce the "bias" dimension in the sentence style embedding through the supervised training with parallel corpus. However, the "bias" dimension is often entangled with other features and can hardly be separated from other dimensions without impairing the fluency or content of text [5, 22]. Recently, some works [25, 31, 41] proposed two-step approaches to improve debiasing quality. These methods firstly identify bias in the text, and then generate modification based on results of the first step. Since the accurate and robust classification of biased texts can be a key step for high-quality automatic text debiasing, our work focuses on biased text classification.

Contrastive Learning. Contrastive learning has gained increased interests among researchers in recent years. Its main idea is to empower the model to pull similar instances closer to each other in the embedding space, while pushing different instances away from each other [13, 53, 59]. Self-supervised contrastive learning has been widely applied in computer vision [4, 23, 34, 35, 50], where data augmentation is utilized to allow the model to learn compact and discriminative features. More recently, many efforts have been taken to extend contrastive loss from the self-supervised setting into the supervised setting [12, 20, 56], where label information is leveraged— instances of the same class are treated as "positive" examples while instances of different classes are treated as "negative" examples. In the NLP domain, supervised contrastive learning is incorporated in pre-training and fine-tuning of language models for a variety of downstream tasks, including machine translation [38], out-of-distribution detection [69], fine-grained classification [49, 62], sentiment analysis [19, 26], metaphor detection [28], and sentence embedding [9, 11, 29, 61]. However, for detecting biased texts, annotations (i.e., labels) can be scarce, noisy, and even biased in themselves. Therefore, utilizing supervised contrastive learning (which only relies on the label information) to differentiate biased and non-biased texts may fall short in performance; new methods are needed to improve the biased text classifier's performance beyond what can be learned solely from the labels. To the best of our knowledge, we are the first to utilize self-supervised contrastive learning to address the problem of learning biased text classifers from scarce, noisy, and biased annotations.

3 METHODOLOGY

In this section, we outline our algorithmic approach for biased language detection.

3.1 Problem Setup

We start by formally defining the biased language classification problem. Given a dataset $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{K}$, where $\boldsymbol{x_i} \in \mathcal{X}$ is the input text and $y_i \in \{0, 1\}$ is the binary label representing whether the input text $\boldsymbol{x_i}$ is biased (1 means it is biased), we aim at learning a model $y = h(\boldsymbol{x})$, which can be used to classify whether any input text $\boldsymbol{x}$ is biased or not. To do so, we typically first need to apply an encoder $f_\theta : \mathcal{X} \rightarrow \mathcal{Z} \in \mathbb{R}^{|\mathcal{X}| \times d}$ to map the input texts into a feature space $\mathcal{Z}$, where θ is the parameters of the encoder and d is the dimension of the feature space. Then, another function (parameterized by ω) $\mathcal{M}_\omega : \mathcal{Z} \rightarrow \mathcal{P}$ will be used to map the encoded features into probabilities in the probability space to indicate the chance for the texts to be biased (e.g., $\mathcal{M}_\omega$ can be a multilayer perceptron). That is, $Pr(y = 1|\boldsymbol{x}) = \mathcal{M}_\omega(f_\theta(\boldsymbol{x}))$. A common approach for learning θ and ω is to search through the parameter space to find the optimal combination of θ and ω that can minimize the

[1]This dataset is publicly available at https://github.com/ZhuoranLu/Bias-detection-annotation.

cross-entropy loss:

$$\ell_{ce}(\theta,\omega) = -\frac{1}{K}\sum_{i=1}^{K} y_i \log(\mathcal{M}_\omega(f_\theta(\boldsymbol{x}))) + (1-y_i)\log(1-\mathcal{M}_\omega(f_\theta(\boldsymbol{x}))) \tag{1}$$

3.2 Enhancing the Encoder with Supervised Contrastive Learning

The quality of the biased language classifier learned through minimizing the cross-entropy loss function largely depends on the quality of the encoder function f_θ. Directly fine-tuning pre-trained text encoders may not be optimal given the subtle semantic differences between some biased and non-biased texts—texts that are close to each other in the pre-trained embedding space may in fact have different labels. To enhance the classifier's ability to distinguish text instances of different classes (i.e., biased, non-biased), an intuitive thought is to utilize the supervised contrastive learning [12, 69] to conduct representation learning, i.e., learn a better encoder. Specifically, given the dataset $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{K}$, the supervised contrastive loss function for learning the optimal parameters θ of the encoder is formulated as:

$$\ell_{scl}(\theta) = -\sum_{i=1}^{K} \frac{1}{K|S(i)|} \sum_{s\in S(i)} \log \frac{\exp(\text{sim}(\boldsymbol{z_i}, \boldsymbol{z_s})/\tau)}{\sum_{n\in N(i)} \exp(\text{sim}(\boldsymbol{z_i}, \boldsymbol{z_n})/\tau)} \tag{2}$$

where $\boldsymbol{z_i} = f_\theta(\boldsymbol{x_i})$ is the embedding feature vector of $\boldsymbol{x_i}$, $S(i)$ is the collection of the index of text instances in $\mathcal{D}$ with the same label as $\boldsymbol{x_i}$ (i.e., the "positive set"), $N(i)$ is the collection of the index of text instances in $\mathcal{D}$ with different labels than $\boldsymbol{x_i}$ (i.e., the "negative set"), $\text{sim}(\cdot,\cdot)$ is the function for measuring the similarity between two instances (e.g., the cosine similarity), and τ is the temperature hyper-parameter for controlling the similarity between negative instances and positive instances. Using this supervised contrastive loss, we are effectively searching for the optimal encoder f_θ that can pull together text instances belonging to the same class, while simultaneously pushing apart instances from different classes in the embedding space.

Finally, we can incorporate the loss function of the encoder into the overall loss function of the classifier to simultaneously learn θ and ω. For example, the final loss function of the classifier with the enhancement from supervised contrastive representation learning can be :

$$\ell_{scl\_enhanced}(\theta,\omega) = \ell_{ce}(\omega,\theta) + \alpha\ell_{scl}(\theta) \tag{3}$$

where $\alpha > 0$ is weight parameter.

3.3 Enhancing the Encoder with Self-supervised Contrastive Learning

Enhancing the encoder with supervised contrastive learning may encounter some limitations when being applied in practice: First, when the ground-truth labels are collected from experts, the size of the training dataset $\mathcal{D}$ is often relatively small due to the high annotation costs of experts. This limited supervision may imply that the supervised contrastive loss function could only bring about a rather small improvement in optimizing f_θ. On the other hand, when the ground-truth labels are collected from the crowd, the labels can be very noisy and sometimes even reflect the crowd annotators' own biases. This, again, would limit the benefits of utilizing supervised contrastive learning to enhance the encoder.

One possible approach to alleviate the problems stated above is to enhance the encoder with *self-supervised* contrastive learning instead. Different from supervised contrastive learning, in which the label information is directly utilized to generate the positive and negative sets of an instance, self-supervised contrastive learning utilizes data augmentation techniques to generate the positive set of an instance, while all other data instances are put in the negative set. The intuition here is that given a text instance $\boldsymbol{x}$, after applying an augmentation operation to it and thus slightly modifying it into $\boldsymbol{x'}$, $\boldsymbol{x'}$ should still be closer to $\boldsymbol{x}$ in the embedding space (hence belong to the positive set of $\boldsymbol{x}$) compared to any other text instances (which belong to the negative set). Following suggestions made by the recent works [33, 42], we consider multiple different types of augmentation operations instead of a single operation to increase the diversity of the augmented data, which is believed to bring about better performance of self-supervised contrastive learning.

More specifically, let $\boldsymbol{O}$ be a set of augmentation operators, where each operator $o \in \boldsymbol{O}$ is a commonly used text augmentation mapping $o : \mathcal{X} \to \mathcal{X}$. For example, $\boldsymbol{O}$ could contain operators like deleting a random word in the text, randomly substituting a word with its synonym, etc. Suppose we will generate T augmented instances for each text instance in our dataset $\mathcal{D}$, and there exists a policy $\boldsymbol{p} \in \mathbb{R}^{|O|}$ that defines the sampling probability distribution for augmentation operations (i.e., p_j represents the probability that operator o_j will be chosen as the augmentation operator). Then, for a text instance $\boldsymbol{x_i}$ in our dataset, we will sample T operators $\{o_i^t\}_{t=1}^{T} \subset \boldsymbol{O}$ based on the sampling policy $\boldsymbol{p}$, and then apply these operators to $\boldsymbol{x_i}$ to generate the T augmented instances $\{\boldsymbol{x_i^{t\prime}}\}_{t=1}^{T}$ where $\boldsymbol{x_i^{t\prime}} = o_i^t(\boldsymbol{x_i})$. The self-supervised contrastive loss function for learning the optimal parameters θ of the encoder is then formulated as:

$$\ell_{sscl}(\theta;\boldsymbol{p}) = -\sum_{i=1}^{K} \frac{1}{KT} \sum_{s\in\{\boldsymbol{x_i^{t\prime}}\}_{t=1}^{T}} \log \frac{\exp(\text{sim}(\boldsymbol{z_i}, \boldsymbol{z_s})/\tau)}{\sum_{n\in\{\mathcal{X}\setminus\boldsymbol{x_i}\}} \exp(\text{sim}(\boldsymbol{z_i}, \boldsymbol{z_n})/\tau)} \tag{4}$$

where $\boldsymbol{z_i} = f_\theta(\boldsymbol{x_i})$ and the same encoder is applied both on the original text instances and the augmented instances. When the sampling policy $\boldsymbol{p}$ of augmentation operations is chosen appropriately, ℓ_{sscl} encourages the encoder to generate more discriminative features by maximizing similarities between augmented views of the same data instance and minimizing similarities between different data instances. Compared to supervised contrastive learning, self-supervised contrastive learning has the potential to address the data scarcity and data noise/biases concerns because (1) the size of the "dataset" that can be used to learn f_θ has increased due to data augmentation, and (2) the encoder is optimized to capture the semantic similarity rather than label similarity, thus is less affected by the noise/biases in the labels.

3.4 Learning the Robust Encoder

We note that the quality of the encoder obtained via self-supervised contrastive learning may highly rely on the quality of the sampling policy $\boldsymbol{p}$. This is because some operators, when applied inappropriately, may change the label for the augmented instances. For example, the operator of "random deletion" may delete the biased word in the input text, which will turn an original biased text instance into

a non-biased augmented instance. However, the self-supervised contrastive loss function still attempts to pull together these two instances in the embedding space, which can be misleading.

In light of the importance of picking the right $\boldsymbol{p}$, in the ideal case, one would hope to use self-supervised contrastive learning to learn f_θ given the optimal $\boldsymbol{p}^*$, which may need to be designed by experts. But what if $\boldsymbol{p}^*$ is not available? Inspired by the ideas in adversarial training [44, 52], we propose that in the absence of the optimal sampling policy, we should learn an encoder f_θ that is *robust*—it should achieve the best performance even if the sampling policy $\boldsymbol{p}$ is chosen to intentionally confuse the encoder. More specifically, we propose to simultaneously learn the encoder parameters θ and the sampling policy $\boldsymbol{p}$ by solving a min-max optimization problem:

$$\begin{aligned} &\min_\theta \ \ell_{sscl}(\theta; \boldsymbol{p}) \\ &\text{s.t. } \boldsymbol{p} \in \operatorname{argmax}_{\boldsymbol{p}}[\ell_{sscl}(\theta; \boldsymbol{p}) - \frac{\gamma}{2}\text{Dist}(\boldsymbol{p}, \boldsymbol{u})] \end{aligned} \tag{5}$$

where $\gamma \in \mathbb{R}$ is a positive coefficient, $\boldsymbol{u}$ is the discrete uniform distribution (i.e., $u_j = \frac{1}{|O|}, \forall j \in [1, |O|]$), and $\text{Dist}(\cdot, \cdot)$ is the Euclidean distance function between two discrete sampling distribution, which serves as a regularizer to avoid policy collapse during the training process [52].

To solve Eq. 5, we optimize the model parameter θ and the sampling policy $\boldsymbol{p}$ alternately. Specifically, when $\boldsymbol{p}$ is fixed, the optimization of encoder model parameters θ can be conducted via gradient descent. However, when θ is fixed, finding the optimal value of $\boldsymbol{p}$ is more difficult. This is because the exact gradient of $\boldsymbol{p}$ is hard to calculate since ℓ_{sscl} is non-differentiable to the sampling policy $\boldsymbol{p}$. As a result, we try to estimate the gradient $\widetilde{\nabla_{\boldsymbol{p}}}$ through a perturbation-based method. Here, we define a small perturbation on as $\Delta\boldsymbol{p} \in \mathbb{R}^{|O|}$, which is uniformly sampled from a unit sphere, and the perturbated policy is $\boldsymbol{p'} = \boldsymbol{p} + \Delta\boldsymbol{p}$. Intuitively, the desired perturbation should increase ℓ_{sscl} so that more challenging operators could be sampled via $\boldsymbol{p'}$. Therefore, we define a signal function S as:

$$\begin{aligned} S(\boldsymbol{p}, \Delta\boldsymbol{p}) =& \ell_{sscl}(\theta; \boldsymbol{p} + \Delta\boldsymbol{p}) - \ell_{sscl}(\theta; \boldsymbol{p}) - \\ & \frac{\gamma}{2}(\text{Dist}(\boldsymbol{p} + \Delta\boldsymbol{p}, \boldsymbol{u}) - \text{Dist}(\boldsymbol{p}, \boldsymbol{u})) \end{aligned} \tag{6}$$

We then approximate the gradient $\widetilde{\nabla_{\boldsymbol{p}}}$ via the Monte Carlo estimate [36]:

$$\widetilde{\nabla_{\boldsymbol{p}}} := \frac{1}{B}\sum_{b=1}^{B} \Phi(\boldsymbol{p}, \Delta\boldsymbol{p}_b)\Delta\boldsymbol{p}_b \tag{7}$$

where $\Phi(\boldsymbol{p}, \Delta\boldsymbol{p}) = \text{sign}(S(\boldsymbol{p}, \Delta\boldsymbol{p}))$, and B is the number of perturbations we sample. Furthermore, to ensure that $\boldsymbol{p}$ is a valid sampling distribution, we constrain the policy via the following projection:

$$\text{proj}(\boldsymbol{p}) = \begin{cases} \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|_1} - \epsilon, & \text{if } \|\boldsymbol{p}\|_1 \neq 1 \\ \boldsymbol{p}, & \text{otherwise} \end{cases} \tag{8}$$

where ϵ is a small constant to ensure numerical stability[2]. The full update for the sampling policy $\boldsymbol{p}$ is:

$$\boldsymbol{p} \leftarrow \text{proj}\left(\boldsymbol{p} + \frac{\lambda}{B}\sum_{b=1}^{B} \Phi(\boldsymbol{p}, \Delta\boldsymbol{p}_b)\Delta\boldsymbol{p}_b\right) \tag{9}$$

[2]In our experiments , we used $\epsilon = 1e-6$.

Algorithm 1: Alternating Optimize (θ, ω), $\boldsymbol{p}$

Input : initial policy $\boldsymbol{p}_0$, model f, $\mathcal{M}$ parameterized by θ_0, ω_0
Output: Updated policy $\boldsymbol{p}$, updated model parameters θ, ω

```
for i ← 1 to N do
    (θ_i, ω_i) ← (θ_{i-1}, ω_{i-1}) − η∇_{θ,ω} ℓ_cleare(θ_{i-1}, ω_{i-1}; p_{i-1})
     // Update θ, ω;
    Estimate ∇̃_p given (θ_i, p_{i-1}) // Eq.7
    p_i ← proj(p_{i-1} + λ∇̃_p)// Eq.9
end
```

where $\lambda > 0$ is the step size.

3.5 Overall Framework

Finally, we propose Contrastive Learning with Robust Encoder (CLEARE) by integrating the robust encoder into the biased language classifier. We do so by solving the following min-max optimization problem:

$$\begin{aligned} &\min_{\theta,\omega} \ \ell_{cleare}(\theta, \omega; \boldsymbol{p}) = \ell_{ce}(\theta, \omega) + \alpha\ell_{scl}(\theta) + \beta\ell_{sscl}(\theta; \boldsymbol{p}) \\ &\text{s.t. } \boldsymbol{p} \in \operatorname{argmax}_{\boldsymbol{p}}[\ell_{sscl}(\theta; \boldsymbol{p}) - \frac{\gamma}{2}\text{Dist}(\boldsymbol{p}, \boldsymbol{u})] \end{aligned} \tag{10}$$

where $\alpha, \beta > 0$ are the weight parameters. Algorithm 1 summarizes the alternating optimization process of this loss function. In particular, in each iteration, we first update parameters θ and ω given the current sampling policy $\boldsymbol{p}$, and then we update the policy following the perturbation-based method to identify a more challenging data augmentation policy.

4 EVALUATION 1: TOWARDS MORE ACCURATE DETECTION

In this section, we present a set of evaluation studies on three existing datasets of online biased language to understand how CLEARE could help improve the accuracy in biased language detection.

4.1 Experimental Settings

4.1.1 Datasets. We considered three annotated datasets of online biased language in this set of evaluations. The statistics of these three datasets are shown in Table 1. The biased text instances in these datasets cover a wide variety of biases, such as biases towards specific gender, race, and political groups.

- **WIKI-Bias** [68]: This is a parallel corpus extracted from the Wikipedia edits. It consists of over 4,000 biased and neutralized sentence pairs labeled manually, with very nuanced differences between each pair of biased and unbiased text.
- **BABE** [47]: This dataset contains 3,700 sentences extracted from online news articles, and trained linguistic experts are recruited to label whether each sentence is biased or not. Thus, the label quality of this dataset is expected to be relatively high.
- **MBIC** [48]: This dataset contains 1700 sentences that are extracted from news articles, and the biased/unbiased label of each sentence is annotated by crowd workers.

4.1.2 Comparison approaches. We compared the performance of our proposed approach, CLEARE, in classifying biased language against the following baselines:

Dataset	Total	Train	Val	Test	len
WikiBias	8198	5028	1066	2104	29.5
BABE	3700	-	-	-	35.5
MBIC	1700	-	-	-	37.5

Table 1: The number of text instances in each dataset. "-" denotes the original dataset doesn't provide train/validation/test splits. len represents the average length of the text.

- **Feature-based methods**: We used the *GloVe* embedding [40] to represent each sentence, and then tuned different supervised models (i.e., SVM, MLP, TextCNN [63]) to classify whether the sentence is biased.
- **Fine-tuning encoder based methods**: We considered two baseline methods based on fine-tuning the pre-trained encoder: (1) CE: the standard cross-entropy loss function (Eqn. 1) is used to simultaneously learn the parameters of the encoder (i.e., θ) and the classifier (i.e., ω). (2) SCLEN: the loss function with enhancement of supervised contrastive learning (Eqn. 3) is used to simultaneously learn the parameters of the encoder and the classifier.
- **Distant supervision Methods**: In previous studies [47, 68], researchers have used more than 100,000 additional training data points to fine-tune the pre-trained encoders and therefore have improved the performance of the biased language classifiers. Since we do not have access to this additional training data, in our comparison, we directly report the performance of these classifiers as stated in [47, 68].

4.1.3 Training details. For both the fine-tuning encoder based baseline methods and CLEARE, we take the BERT model [7] and the ROBERTa [30] model from Huggingface's transformers library [55] as our pre-trained encoders. We then used the representation embeddings of the last layer of the pre-trained models as the input of the classifier (i.e., $\mathcal{M}_\omega$). The classifier consists of one hidden layer with 768 nodes and one output layer.

For data augmentation in CLEARE, $\mathcal{O}$ consists of five operations—Random Swap (RS), Random Delete (RD), Synonym Substitute (SS), TF-IDF and Contextual Sentence Insert (CSI). For more details about augmentation operations, see Appendix B.

For all methods, we performed a hyper-parameter search on the validation set over initial learning rate, weight decay, dropout rate, etc. Following previous studies [28, 69], we empirically set $\tau = 0.2$ (Eqn. 2 and 4), $\gamma = 0.1$ (Eqn. 5), $B = 2000$ (Eqn. 7), $\lambda = 0.01$ (Eqn. 9), $\alpha = 2$ (Eqn. 3 and 10) and $\beta = 0.5$ (Eqn 10). Except for *GloVe*+SVM, all methods are optimized with Adam [21] with an initial learning rate of $5e^{-5}$ and a batchsize of each training iteration of 32. We fine-tuned all fine-tuning encoder based methods for 20 epochs.

Finally, when evaluating on the WIKI-Bias dataset, we directly used their training, validation, and test splits. For each method (except for the distant supervision methods), we repeated it five times with different parameter initialization, and then reported the average performance score of it. On the other hand, when evaluating on the BABE and MBIC datasets, we randomly split the dataset into three partitions with 70%, 5%, and 25% of the entire dataset, and used them as our training, validation and test sets, respectively. We repeated this process five times, and the average performance scores of different methods across these five repetitions were then reported, except for the distant supervision methods. Following previous works [47], we evaluated the performance of the classifiers trained using different methods with Macro-F1 and AUROC.

4.2 Evaluation Results

4.2.1 Comparisons with baselines. Table 2 presents the comparison in the biased language classifiers' performance when trained using different methods. Overall, we observed that the proposed method, CLEARE, consistently outperforms the best baseline methods on all three datasets. Below, we summarized a few key observations from the comparison.

Fine-tuning encoder based methods are better than feature-based methods. We found that methods that are based on fine-tuning pre-trained language models (e.g., BERT + CE, BERT + SCLEN, and our method), in general, outperform feature-based methods like *GloVe* + TextCNN. We attribute such improvement to self-attention mechanism of transformers, which can more efficiently capture the subtle semantic differences between texts.

Contrastive self-training with robust encoder is better than solely supervised training. We compared CLEARE with methods that only utilize supervised training signals, including both CE and SCLEN. Again, CLEARE almost always outperforms these two methods on the three datasets, both when the pre-trained model adopted is BERT and ROBERTa. To better understand why CLEARE achieves a better performance than training methods that only utilize the direct supervisions, we used t-SNE[3] [51] to visualize the feature space of different methods when we built classifiers for the WIKI-Bias dataset, and ROBERTa was used as the pre-trained encoder model[4]. Figure 1 shows the visualization results. In the CE or SCLEN space, non-biased and biased text instances are entangled and mixed, which suggests that CE and SCLEN may fall short in capturing the subtle semantic information in the challenging WIKI-Bias dataset, where only very nuanced differences exist between non-biased and biased texts. In contrast, in the CLEARE space, despite some noise, most of the text instances with different labels are disentangled while the clusters with different labels also become more compact. This demonstrates that CLEARE motivates the model to learn more discriminative feature representations for classification.

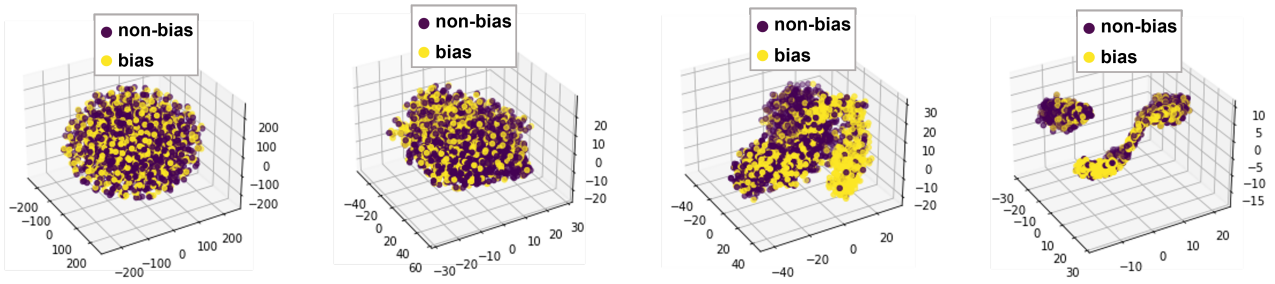
Contrastive self-training with robust encoder is on par with distant supervision methods. Finally, we found that CLEARE achieved similar or even better performance than distant supervision methods. Consider that the distance supervision methods are trained based on a large volume of additional training data, the capability of CLEARE to achieve a comparable performance using a much smaller amount of data is notable.

4.2.2 Understand the advantage of the robust encoder. CLEARE is designed to learn a robust encoder that can achieve good classification performance even if the data augmentation sampling policy (i.e., $\boldsymbol{p}$) is chosen adversarially. To further understand the advantage

[3]The perplexity parameter was set as 40 across all methods.
[4]Compared with the BABE and MBIC dataset, the semantic difference between non-biased and biased text in the WIKI-Bias dataset is more subtle, which places a higher requirement for the model's discriminative ability.

Dataset	WIKI-Bias		BABE		MBIC	
	Macro-F1	AUROC	Macro-F1	AUROC	Macro-F1	AUROC
GloVe + SVM	22.3	51.2	60.4	67.2	65.1	58.4
GloVe + MLP	25.7	52.3	59.2	66.8	68.7	60.4
GloVe + TextCNN	42.8	56.5	70.2	78.1	65.8	70.9
BERT + CE	54.7	68.4	78.7	86.7	76.3	83.1
ROBERTa + CE	53.4	66.7	79.6	86.6	77.4	85.3
BERT + SCLEN	61.8	71.9	78.5	86.6	77.4	83.9
ROBERTa + SCLEN	64.4	72.7	80.1	87.5	78.1	85.9
BERT + distant	65.8	-	80.4	-	77.8	-
ROBERTa + distant	-	-	79.9	-	79.8	-
BERT + CLEARE	67.5	74.5	79.9	87.5	78.2	84.4
ROBERTa + CLEARE	63.2	73.2	81.3	88.5	79.4	87.1

Table 2: Comparing the performance of CLEARE with baseline methods on 3 datasets, in terms of Macro-F1 (%) and AUROC (%). The best method in each column is colored in blue and second best is colored in light blue. "-" denotes the result is not available. All results except "*distant*" are averaged over 5 runs.



(a) w/o fine-tuning **(b) CE** **(c) SCLEN** **(d) CLEARE**

Figure 1: 3D t-SNE visualization of feature spaces before fine-tuning, and after applying three different fine-tuning methods on the WIKI-Bias dataset.

of this robust encoder, we conducted an experiment to compare the performance of CLEARE with a few other contrastive self-training methods with fixed sampling policy (i.e., we directly optimize for the objective function in Eqn. 10 with a fixed $\boldsymbol{p}$). Specifically, we considered seven baseline data augmentation operation sampling policies, including 5 single-operator policies (i.e., only a single type of operator is used in data augmentation) and 2 multiple-operators policies (i.e., multiple types of operators is used in data augmentation). For single-operator policies, we considered applying only the operator of Random Swap (RS), Random Delete (RD), TF-IDF, Synonym Substitute (SS), and Contextual Sentence Insert (CSI). For multiple-operator policies, we adopted two baseline methods proposed in the existing literature: EDA [54] and JOAO [58] (For more details about these baseline policies, see Appendix B).

We followed the same experimental settings as described in Section 4.1.3 to train different models, and then compared the performance of CLEARE with the seven baseline methods. Since results in Table 2 suggest that models trained using the ROBERTa encoder seems to have similar or sometimes better performance compared to models trained using the BERT encoder, in this experiment, all our models are trained using only the ROBERTa model as the pre-trained encoder. Table 3 reports our comparison results. We first note that none of models utilizing the single-operator sampling policies achieve the best classification performance across all three datasets. This suggests the choice of the "optimal" sampling policy is highly context-dependent. Furthermore, we find the CLEARE almost always consistently outperform other models, including both models trained with single-operator policies and models trained with multiple-operator policies. This suggests the advantage of CLEARE in learning a robust encoder over other models that are trained based on a fixed, and possibly non-optimal, sampling policy.

4.2.3 Robustness Analysis. Finally, we conducted two simulated evaluations to understand how robust CLEARE is with respect to the size of the training data and the level of noise in labels. We chose to use the BABE dataset in this simulated evaluations as we expect the labels for the BABE dataset is of high quality (since they are provided by linguistic experts). We focused on the comparison between the three models where ROBERTa was used as the pre-trained encoder, while either CE, SCLEN, or CLEARE was used for fine-tuning the encoder and learning the parameters of the classifier.

In our first simulation, to simulate different degrees of labeled data scarcity, we varied the size of the data used for training the models from 25%, 50%, 75% to 100% of the entire training dataset. Again, for each of the three models (i.e., ROBERTa + CE, ROBERTa + SCLEN, ROBERTa + CLEARE), we trained it for 5 runs, each run with different random parameter initializations, and Figure 2a

Methods		RS	RD	TFIDF	SS	CSI	EDA	JOAO	CLEARE
WIKI-Bias	**Macro-F1**	67.1	64.2	66.3	66.8	66.7	66.4	66.4	67.5
	AUROC	74.2	72.1	73.9	73.6	72.9	74.0	74.0	74.5
BABE	**Macro-F1**	78.8	79.7	81.1	80.8	81.6	80.9	80.9	81.3
	AUROC	86.4	87.9	87.7	87.2	88.3	87.9	87.9	88.5
MBIC	**Macro-F1**	78.4	76.1	78.8	76.8	78.5	77.8	77.8	79.4
	AUROC	86.4	84.4	86.3	85.4	86.4	85.8	85.8	87.1

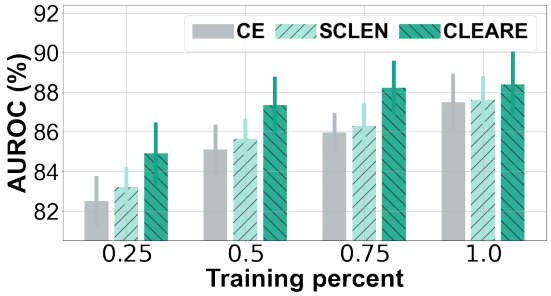
Table 3: Comparing the performance of CLEARE with different augmentation strategies on 3 datasets, in terms of Macro-F1 (%) and AUROC (%). The best method in each row is colored in blue and second best is colored in light blue. All results are averaged over 5 runs, with ROBERTa being used as the pre-trained encoder model.

Dataset	left-leaning		right-leaning		center		-		Total	
	biased	non-biased	biased	non-biased	biased	non-biased	biased	non-biased	biased	non-biased
BABE	630	381	598	394	99	592	509	497	1836	1864

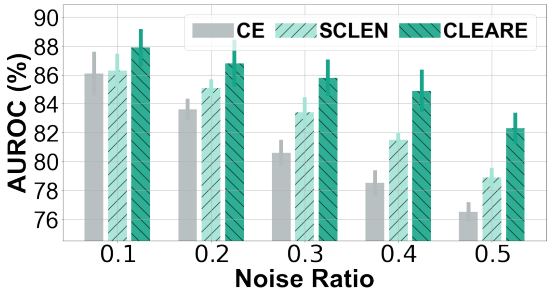
Table 4: Detailed statistics of the BABE dataset. "-" denotes the political leaning of text is controversial and thus not decided.

shows the performance comparisons across these three models. It is clear from the figure that as the size of the training data decreases, the performance of all three models degrades. However, we found that CLEARE consistently outperforms the other two models on all training datasets with different sizes, and it has a relatively mild performance degradation. For example. When only 50% of the labeled training data is available, the drop of AUROC for CLEARE compared to the case when the model is trained on the entire training dataset is only 0.6%, while the drop for CE and SCL are 2.6% and 2.2%, respectively. These observations suggest that utilizing contrastive self-supervised learning with a robust encoder, CLEARE exhibits the potential to alleviate the concerns raised by the scarcity of labeled data in biased language detection.

In our second simulation, the models are trained with full training data, but we randomly added noise into the labels in the training dataset. Specifically, to simulate different levels of noise in the training data, we constructed five simulated training datasets by randomly choosing 10%, 20%, 30%, 40%, or 50% of the data in the original training dataset and flipping their labels. The average performance of the three models across 5 runs with random parameter initialization is reported in Figure 2b. Here, we observed that the performance of the three models degrades when the noise level becomes higher, but CLEARE still outperforms the other two models consistently, and its advantage over the two baseline models become particularly salient when the noise level in the training dataset is high. These results, thus, demonstrate the robustness of CLEARE against the noisy annotations of biased language.



(a) Performance comparison w.r.t. training set size.



(b) Performance comparison w.r.t. label noise level.

Figure 2: Comparing the performance of CLEARE with baseline methods when changing the size of training data or varying the level of label noise. Error bars represent the standard errors of the mean.

5 EVALUATION 2: TOWARDS FAIRER DETECTION

In this section, we explore the potential of CLEARE for advancing fairer detection of biased language. In particular, as annotators may often inject their own biases (e.g., confirmation bias) into the labels they provide on different text instances when determining whether they are biased [10], biased language classifiers built on top of these annotations may suffer from a fairness issue—For example, if the majority of annotators hold liberal views, then the classifiers trained based on data labeled by them may tend to classify texts that align with liberal views as non-biased and texts that align with conservative views as biased. We conjecture that since the training of CLEARE utilizes the semantic information embedded in the text beyond the direct supervision from the (potentially biased) labels, it may result in fairer classifiers.

To validate this conjecture, we conducted two additional evaluation studies. In these evaluations, we focus on the BABE dataset, since for a subset of the sentences in the BABE dataset, experts also provide the label on the sentence's political leaning (i.e., left-leaning, right-leaning and center; see Table 4 for the statistics). This enables us both to construct simulated dataset to reflect a population of annotators with different compositions in their political leanings,

and to collect labels from real-world human annotators to examine whether they exhibit biases in their labels and how fair different methods are when being trained on the real-world labeled data.

5.1 Evaluation Metrics for Fairness

Following previous work [8], we measured the fairness level of a classifier using *error rate equality difference*, which is the difference between error rate across different "groups". For example, consider false positive rate (FPR) and false negative rate (FNR) as the two metrics to quantify error rates, we can define two different versions of error rate equality difference—the *false positive equality difference* (FPED) and the *false negative equality difference* (FNED):

$$FPED = \sum_{s \in S} |FPR - FPR_s| \tag{11}$$

$$FNED = \sum_{s \in S} |FNR - FNR_s| \tag{12}$$

where FPR and FNR are false positive rate and false negative rate on the entire testset S, while FPR_s and FNR_s are the FPR and FNR on a specific subset s of S. In our evaluation, s represents the subsets containing texts with the same political leaning. For simplicity, we consider only the subset of left-leaning text and the subset of right-leaning text in computing FPED and FNED. Intuitively, the lower FPED and FNED are, the smaller the classifiers' performance difference on texts with different political leanings, and the "fairer" the classifier.

5.2 Simulation Study

We firstly conducted a simulation study by generating two simulated training datasets based on BABE to reflect how two populations of annotators with different composition of political leanings may label the same data, assuming that a subset of the annotators suffer from their confirmation biases in labeling the text (e.g., a left-leaning annotator tends to label a biased left-leaning text as non-biased, while labeling a non-biased right-leaning text as biased).

- **left-dominant dataset**: In this dataset, we aim to simulate that the data is primarily labeled by a set of left-leaning annotators. We thus changed the label of 50% of the biased, left-leaning text in the training data into "non-biased," and changed the label of 50% of the non-biased, right-leaning text in the training data into "biased."
- **right-dominant dataset**: In this dataset, we aim to simulate that the data is primarily labeled by a set of right-leaning annotators. We thus changed the label of 50% of the biased, right-leaning text in the training data into "non-biased," and changed the label of 50% of the non-biased, left-leaning text in the training data into "biased."

We used ROBERTa as the pre-trained encoder and then fine-tuned it with CE, SCLEN and CLEARE, respectively. For data split, we randomly chose 70% of the **left-leaning** and **right-leaning** text, as well as 10% of the **center** text and text without political leaning labels (i.e., the - subset) as the training set, and 5% of the data in all subsets as the validation set. All the remaining data was then used as the test set. We trained and tested the three models given this data partition, and then repeated this process five times over different random partition of the dataset.

Methods	left-leaning		right-leaning		*FPED* ↓	*FNED* ↓	Total Acc ↑
	FPR ↓	*FNR* ↓	*FPR* ↓	*FNR* ↓			
CE	23.1	35.3	35.2	21.9	12.1	14.4	77.3
SCLEN	24.1	30.6	34.3	**18.5**	10.5	13.5	78.1
CLEARE	**18.6**	**25.7**	**26.8**	19.8	**7.1**	**7.0**	**79.3**

Table 5: Comparing the performance of CLEARE with CE and SCLEN on the left-dominant dataset; left-leaning and right-leaning represent subsets of the test set containing left-leaning and right-leaning text, respectively. Total Acc reports the overall accuracy on all test data. ↓ denotes the lower the better, ↑ denotes the higher the better. Best results are highlighted in bold. All results are averaged over 5 runs.

Methods	left-leaning		right-leaning		*FPED* ↓	*FNED* ↓	Total Acc ↑
	FPR ↓	*FNR* ↓	*FPR* ↓	*FNR* ↓			
CE	31.8	23.6	**16.1**	38.3	15.6	14.7	76.5
SCLEN	25.2	22.5	18.1	37.6	7.1	15.1	77.6
CLEARE	**24.1**	**16.1**	17.9	**26.1**	**6.2**	**8.8**	**79.7**

Table 6: Comparing the performance of CLEARE with CE and SCLEN on the right-dominant dataset. ↓ denotes the lower the better, ↑ denotes the higher the better. Best results are highlighted in bold. All results are averaged over 5 runs.

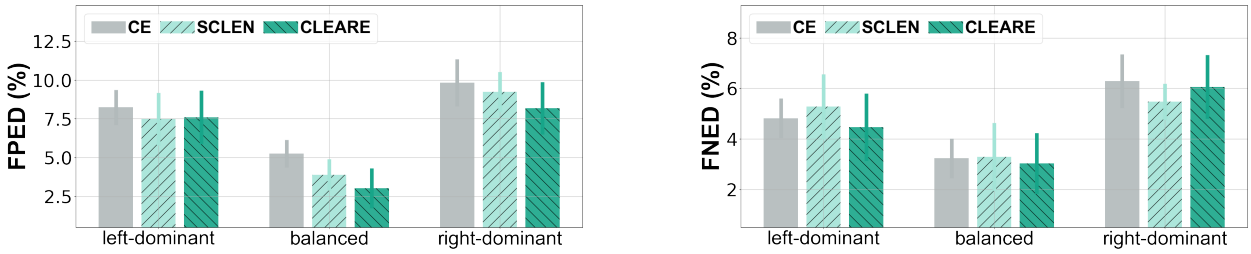
Table 5 and Table 6 report the comparison results of CE, SCLEN and CLEARE on the left-dominant and right-dominant datasets, respectively. We found that compared to CE and SCLEN, CLEARE has a consistently lower FPED and FNED, suggesting it results in a fairer classification across left-leaning and right-leaning text. In the meantime, the accuracy of CLEARE is also higher than the other two baseline methods, suggesting that in this simulation study, the improvement of CLEARE does not come with the decrease in model accuracy.

5.3 Real-World Study

To further understand the performance of CLEARE in the real world scenarios, in which annotators may be subject to many different types of biases in their annotation, we conducted a large-scale crowdsourced data collection with real human annotators on Amazon Mechanical Turk (MTurk), and investigated the performance of our proposed method against the baseline methods on this real-world dataset.

5.3.1 Crowdsourced Annotation Collection. We posted a human intelligence task (HIT) on MTurk to U.S. workers only, and the recruited workers were asked to annotate whether sentences taken from the BABE dataset are biased or not. Specifically, upon arrival at the HIT, we gave workers a brief instruction on how to determine whether a piece of text is biased. Then, workers were asked to complete 20 tasks to review 20 sentences and judge whether the sentence was biased or unbiased[5]. Before submitting the HIT, workers were also asked to self-report their political party affiliation ("Independent", "Democrat," or "Republican"). To filter out potential spammers, We randomly placed an attention check question in our HIT, in which workers were asked to choose a pre-specified

[5]For simplicity, we only asked workers to annotate left-leaning, right-leaning, and center text in the BABE dataset.



(a) *FPED* w.r.t. three compositions of annotators.

(b) *FNED* w.r.t. three compositions of annotators.

Figure 3: Comparing the fairness of CLEARE with baseline methods when the annotated dataset come from different populations of annotators with different composition of political stance. Error bars represent the standard errors of the mean.

Text \ Annotators		Democrat	Republican	Significance
left-leaning	biased	0.61	0.76	$p < 0.001 * **$
	non-biased	0.39	0.29	$p < 0.001 * **$
right-leaning	biased	0.68	0.55	$p < 0.001 * **$
	non-biased	0.36	0.40	$p < 0.01 * *$
Overall accuracy		0.54	0.53	$p > 0.05$

Table 7: Average annotation accuracy on subgroups of BABE texts by annotators who self-report to be Democrat or Republican.

answer. In addition, to encourage crowd workers to try their best in detecting the biased text, we told workers if their overall accuracy in the HIT is over 65%, they could earn an additional bonus of 5 cents for each correct judgment they made in the HIT.

After filtering out those workers who answered the attention check question incorrectly or workers who reported to be "Independent", in total, we collected 9,832 judgments from 402 crowd workers. Among them, 276 workers reported themselves to be Democrat and 126 workers are Republican.

5.3.2 Annotation Results. We firstly set out to understand the annotation accuracy difference across all tasks between the Democrat and Republican workers. As shown in Table 7, we observed that there is no significant difference in overall annotation accuracy for workers with different political stances ($p > 0.05$). However, by taking a deeper look into people's annotation performance on different subgroups of texts, we found that annotators' political stance significantly affected their labeling results. For instance, for left-leaning, biased texts, Democrat workers show a significantly lower accuracy than Republican workers, and our t-test result suggests the difference is significant ($p < 0.001$). On the contrary, on the right-leaning, biased texts, Democrat workers have substantially higher accuracy than Republican workers ($p < 0.001$). In other words, the accuracy differences between Democrat and Republican workers on texts with different political leaning suggest that real-world crowd workers indeed suffer from their own biases when determining whether texts contain biased language.

From the findings above, it is reasonable to conjecture that *left*-leaning workers suffer from their own confirmation bias and have a higher tendency to incorrectly classify right-unbiased text into the biased. At the same time, *right*-leaning workers also prefer to claim that left-unbiased data is biased. In general, people are more tolerant to text with the same political leaning as their stance but more strict to oppositely leaning text data.

5.3.3 Examining the Classifiers' Performance. To examine different biased language classifiers' performance on the real-world dataset, in this experiment, we randomly sample the labeled data with different proportions from the Democrat and Republican workers to create training sets that are labeled by annotator populations with different compositions of political leanings. This allows us to systematically examine the fairness level of different biased language classifiers, as the real-world dataset gets collected from different populations of annotators.

We again focused on comparing the performance of the three methods—CE, SCLEN, and CLEARE, when ROBERTa was used as the pre-trained encoder. We randomly chose 70% of the left-leaning and right-leaning texts, as well as 15% of the "center" text in BABE as the training set. We then took 5% of the texts in all subsets of BABE as the validation set, while the remaining texts were used as the test set. Given the training set, we then constructed three real-world annotated datasets that could be collected from different populations of annotators: (1) *left-dominant*: for each text in the training set, the annotation is sampled from a random Democrat worker; (2) *balanced*: for each text in the training set, with 50% chance the annotation is sampled from a random Democrat worker and with 50% the annotation is sampled from a random Republican worker. (3) *right-dominant*: for each text in the training set, the annotation is sampled from a random Republican worker. For the validation set and test set, the expert labels of the BABE dataset are used for a fair comparison. For each version of the real-world annotated datasets that we constructed, the three models are trained and evaluated. We repeated this process for 30 times over different random partitions of the dataset.

Figure 3 compares the fairness of the three classifiers on the three real-world annotated datasets. We observed a robust and consistent improvement of CLEARE over CE and SCLEN in fairness metrics for all three annotated datasets. Interestingly, we find when the annotations are provided by a balanced population of Democrat and Republican workers (i.e., the "balanced" dataset), all three models exhibit the highest levels of fairness across subsets of texts with different political leanings. Still, CLEARE exhibits significant improvement over the other two baseline methods with respect to

both fairness metrics, which suggests the power of CLEARE for enhancing the fairness level in biased language detection.

6 CONCLUSION

In this paper, we presented our novel method, CLEARE, for biased language detection, where it learns a robust encoder of the textual data via utilizing self-supervised contrastive learning. Our experiments on both real-world datasets and simulations suggest that CLEARE has the potential to address the issues of data scarcity, data noise, and data biases, which are prevalent in real-world application scenarios of biased language detection. We hope that our study, as well as the biased language annotation dataset that we collected from crowd workers with different political leanings in this study, could encourage more work in the future on developing new methods to detect biased language more accurately and fairly.

REFERENCES

[1] Madhusudhan Aithal and Chenhao Tan. 2021. On Positivity Bias in Negative Reviews. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Online, 294–304. https://doi.org/10.18653/v1/2021.acl-short.39

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *NeurIPS*.

[3] Rebecca F Bruce and Janyce M Wiebe. 1999. Recognizing subjectivity: a case study in manual tagging. *Natural Language Engineering* 5, 2 (1999), 187–205.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[5] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. *arXiv preprint arXiv:1905.05621* (2019).

[6] Sunipa Dev, Tao Li, Jeff M. Phillips, and Vivek Srikumar. 2021. OSCaR: Orthogonal Subspace Correction and Rectification of Biases in Word Embeddings. In *EMNLP (1)*. Association for Computational Linguistics, 5034–5050.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

[8] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 67–73.

[9] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP (1)*. Association for Computational Linguistics, 6894–6910.

[10] Meric Altug Gemalmaz and Ming Yin. 2021. Accounting for Confirmation Bias in Crowdsourced Label Aggregation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*.

[11] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. 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)*. Association for Computational Linguistics, Online, 879–895. https://doi.org/10.18653/v1/2021.acl-long.72

[12] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. *In International Conference for Learning Representations* (2021).

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9729–9738.

[14] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*. PMLR, 1587–1596.

[15] Christoph Hube and Besnik Fetahu. 2019. Neural based statement classification for biased language. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 195–203.

[16] Thorsten Joachims. 1997. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In *ICML*. Morgan Kaufmann, 143–151.

[17] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled Representation Learning for Non-Parallel Text Style Transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 424–434. https://doi.org/10.18653/v1/P19-1041

[18] Jussi Karlgren and Douglass Cutting. 1994. Recognizing text genres with simple metrics using discriminant analysis. *In Proceedings of the 15th International Conference on Computational Linguistics* (1994).

[19] Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. 2021. CLASSIC: Continual and Contrastive Learning of Aspect Sentiment Classification Tasks. In *EMNLP (1)*. Association for Computational Linguistics, 6871–6883.

[20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems* (2020).

[21] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

[22] Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

[23] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. 2020. Contrastive representation learning: A framework and review. *IEEE Access* (2020).

[24] Tien-Tsung Lee. 2005. The liberal media myth revisited: An examination of factors influencing perceptions of media bias. *Journal of Broadcasting & Electronic Media* 49, 1 (2005), 43–64.

[25] Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, Retrieve, Generate: a Simple Approach to Sentiment and Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1865–1874. https://doi.org/10.18653/v1/N18-1169

[26] Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021. Learning Implicit Sentiment in Aspect-based Sentiment Analysis with Supervised Contrastive Pre-Training. In *EMNLP (1)*. Association for Computational Linguistics, 246–256.

[27] Chenghua Lin, Yulan He, and Richard Everson. 2011. Sentence subjectivity detection with weakly-supervised learning. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. 1153–1161.

[28] Zhenxi Lin, Qianli Ma, Jiangyue Yan, and Jieyu Chen. 2021. CATE: A Contrastive Pre-trained Model for Metaphor Detection with Semi-supervised Learning. In *EMNLP (1)*. Association for Computational Linguistics, 3888–3898.

[29] Che Liu, Rui Wang, Jinghua Liu, Jian Sun, Fei Huang, and Luo Si. 2021. DialogueCSE: Dialogue-based Contrastive Learning of Sentence Embeddings. In *EMNLP (1)*. Association for Computational Linguistics, 2396–2406.

[30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[31] Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabás Póczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W. Black, and Shrimai Prabhumoye. 2020. Politeness Transfer: A Tag and Generate Approach. In *ACL*. Association for Computational Linguistics, 1869–1881.

[32] Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W Black. 2019. Black is to Criminal as Caucasian is to Police: Detecting and Removing Multiclass Bias in Word Embeddings. *arXiv preprint arXiv:1904.04047* (2019).

[33] Zhengjie Miao, Yuliang Li, and Xiaolan Wang. 2021. Rotom: A Meta-Learned Data Augmentation Framework for Entity Matching, Data Cleaning, Text Classification, and Beyond. In *Proceedings of the 2021 International Conference on Management of Data*. 1303–1316.

[34] Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6707–6717.

[35] Sangwoo Mo, Hyunwoo Kang, Kihyuk Sohn, Chun-Liang Li, and Jinwoo Shin. 2021. Object-aware contrastive learning for debiased scene representation. *Advances in Neural Information Processing Systems* 34 (2021).

[36] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. 2020. Monte Carlo Gradient Estimation in Machine Learning. *J. Mach. Learn. Res.* 21, 132 (2020), 1–62.

[37] Fred Morstatter, Liang Wu, Uraz Yavanoglu, Stephen R Corman, and Huan Liu. 2018. Identifying framing bias in online news. *ACM Transactions on Social Computing* 1, 2 (2018), 1–18.

[38] Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive Learning for Many-to-many Multilingual Neural Machine Translation. 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)*. Association for Computational Linguistics, Online, 244–258. https://doi.org/10.18653/v1/2021.acl-long.21

[39] Kartikey Pant, Tanvi Dadu, and Radhika Mamidi. 2020. Towards detection of subjective bias using contextualized word embeddings. In *Companion Proceedings of the Web Conference 2020*. 75–76.

[40] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[41] Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 34. 480–489.

[42] Shuhuai Ren, Jinchao Zhang, Lei Li, Xu Sun, and Jie Zhou. 2021. Text AutoAugment: Learning Compositional Augmentation Policy for Text Classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 9029–9043. https://doi.org/10.18653/v1/2021.emnlp-main.711

[43] Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*. 105–112.

[44] Alexander Robey, Hamed Hassani, and George J Pappas. 2020. Model-based robust deep learning. *arXiv preprint arXiv:2005.10247* (2020), 2.

[45] Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2021. Societal Biases in Language Generation: Progress and Challenges. 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)*. Association for Computational Linguistics, Online, 4275–4293. https://doi.org/10.18653/v1/2021.acl-long.330

[46] Manjira Sinha and Tirthankar Dasgupta. 2021. Determining Subjective Bias in Text through Linguistically Informed Transformer based Multi-Task Network. In *CIKM*. ACM, 3418–3422.

[47] Timo Spinde, Manuel Plank, Jan-David Krieger, Terry Ruas, Bela Gipp, and Akiko Aizawa. 2021. Neural Media Bias Detection Using Distant Supervision With BABE-Bias Annotations By Experts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 1166–1177.

[48] Timo Spinde, Lada Rudnitckaia, Kanishka Sinha, Felix Hamborg, Bela Gipp, and Karsten Donnay. 2021. MBIC–A Media Bias Annotation Dataset Including Annotator Characteristics. *arXiv preprint arXiv:2105.11910* (2021).

[49] Varsha Suresh and Desmond C. Ong. 2021. Not All Negatives are Equal: Label-Aware Contrastive Loss for Fine-grained Text Classification. In *EMNLP (1)*. Association for Computational Linguistics, 4381–4394.

[50] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. 2020. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *In 34th Conference on Neural Information Processing Systems (NeurIPS) 2020* (2020).

[51] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[52] Jingkang Wang, Tianyun Zhang, Sijia Liu, Pin-Yu Chen, Jiacen Xu, Makan Fardad, and Bo Li. 2021. Adversarial attack generation empowered by min-max optimization. *Advances in Neural Information Processing Systems* 34 (2021).

[53] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[54] Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 6382–6388. https://doi.org/10.18653/v1/D19-1670

[55] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).

[56] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. VideoCLIP: Contrastive Pre-training for Zero-shot Video-Text Understanding. In *EMNLP (1)*. Association for Computational Linguistics, 6787–6800.

[57] Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2000–2010.

[58] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. 2021. Graph Contrastive Learning Automated. *arXiv preprint arXiv:2106.07594* (2021).

[59] Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021. Fine-Tuning Pre-trained Language Model with Weak Supervision: A Contrastive-Regularized Self-Training Approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 1063–1077. https://doi.org/10.18653/v1/2021.naacl-main.84

[60] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 1415–1420. https://doi.org/10.18653/v1/N19-1144

[61] Dejiao Zhang, Shang-Wen Li, Wei Xiao, Henghui Zhu, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021. Pairwise Supervised Contrastive Learning of Sentence Representations. In *EMNLP (1)*. Association for Computational Linguistics, 5786–5798.

[62] Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip S. Yu. 2021. Few-Shot Intent Detection via Contrastive Pre-Training and Fine-Tuning. In *EMNLP (1)*. Association for Computational Linguistics, 1906–1912.

[63] Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* (2015).

[64] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender Bias in Contextualized Word Embeddings. In *NAACL-HLT (1)*. Association for Computational Linguistics, 629–634.

[65] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using

Corpus-level Constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2941–2951. https://www.aclweb.org/anthology/D17-1319

[66] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 15–20. https://doi.org/10.18653/v1/N18-2003

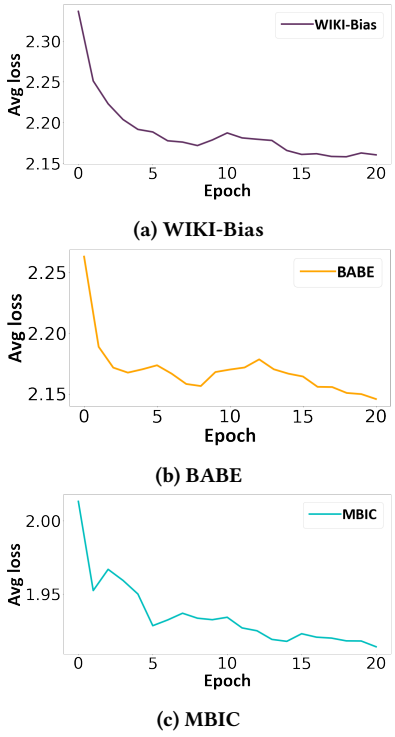
[67] Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning Gender-Neutral Word Embeddings. In *EMNLP*. Association for Computational Linguistics, 4847–4853.

[68] Yang Zhong, Jingfeng Yang, Wei Xu, and Diyi Yang. 2021. WIKIBIAS: Detecting Multi-Span Subjective Biases in Language. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 1799–1814.

[69] Wenxuan Zhou, Fangyu Liu, and Muhao Chen. 2021. Contrastive Out-of-Distribution Detection for Pretrained Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 1100–1111. https://doi.org/10.18653/v1/2021.emnlp-main.84

Appendix A EMPIRICAL CONVERGENCE OF CLEARE

We used Monte Carlo estimate to approximate the gradient for policy $\boldsymbol{p}$ in ℓ_{sscl} (Eqn.5). Figure A1 shows some level of empirical convergence of the optimization process.



(a) WIKI-Bias

(b) BABE

(c) MBIC

Figure A1: Empirical training curves of the loss function value for the three datasets.

Appendix B COMPARISONS WITH DIFFERENT AUGMENTATION METHODS

We extensively compared our method with seven other algorithms using different data augmentation strategies, which consists of single-operation and multiple-operations data augmentation methods. For single-operation methods, we consider:

- Random Swap (RS): RS randomly exchanges the position of adjacent words in the sentence.
- Random Delete (RD): RD randomly deletes words or phrases in the sentence.
- Synonym Substitute (SS): SS randomly replaces words or phrases with their synonyms.
- TF-IDF [16]: TF-IDF utilizes the term frequency and the inverse document frequency to compute the score for each word then replace words with low scores according to proportion.
- Contextual Sentence Insert (CSI): We utilized GPT3 [2] to firstly encode the text inputs into feature space and find a

sentence that has similar semantic features. Then we inserted this new sentence in the original text inputs.

For Random Swap (RS), Random Delete (RD), TF-IDF and Synonym Substitute (SS), we set the proportion p of word to be edited as 0.2 in all experiments. For multiple-operations methods, we consider EDA and JOAO:

- EDA (easy data augmentation) [54]: EDA samples one of five single-operation methods uniformly during the training phase for training data.
- JOAO [58]: JOAO is an adaptive data augmentation method[6] for graph-like data, which applies bi-section optimization to select optimal sampling policy. We replaced graph augmentation operations with the five text augmentation operations described above.

[6]We noticed that TAA [42] was recently proposed as a learnable data augmentation paradigm for text. We did not compare it with JOAO and our method since TAA needs extra validation data to optimize the policy, and it can not be applied in the contrastive learning framework.