Reducing Sentiment Bias in Pre-trained Sentiment Classification via Adaptive Gumbel Attack

Jiachen Tian, Shizhan Chen, Xiaowang Zhang*, Xin Wang, Zhiyong Feng

College of Intelligence and Computing, Tianjin University, Tianjin, China
Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China
{jiachen6677, shizhan, xiaowangzhang, wangx, zyfeng}@tju.edu.cn

Abstract

Pre-trained language models (PLMs) have recently enabled rapid progress on sentiment classification under the pre-train and fine-tune paradigm, where the fine-tuning phase aims to transfer the factual knowledge learned by PLMs to sentiment classification. However, current fine-tuning methods ignore the risk that PLMs cause the problem of *sentiment bias*, that is, PLMs tend to inject positive or negative sentiment from the contextual information of certain entities (or aspects) into their word embeddings, leading them to establish spurious correlations with labels. In this paper, we propose an adaptive Gumbel-attacked classifier that immunes sentiment bias from an adversarial-attack perspective. Due to the complexity and diversity of sentiment bias, we construct multiple Gumbel-attack expert networks to generate various noises from mixed Gumbel distribution constrained by mutual information minimization, and design an adaptive training framework to synthesize complex noise by confidence-guided controlling the number of expert networks. Finally, we capture these noises that effectively simulate sentiment bias based on the feedback of the classifier, and then propose a multi-channel parameter updating algorithm to strengthen the classifier to recognize these noises by fusing the parameters between the classifier and each expert network. Experimental results illustrate that our method significantly reduced sentiment bias and improved the performance of sentiment classification.

Introduction

Sentiment classification (SC) aims to automatically detect the sentiment polarity of an opinion, e.g. positive, neutral, or negative (Phan and Ogunbona 2020). An opinion can be defined as a tuple `<Target, Context>`, where the target refers to any entity or aspect of the entity on which the opinion has been expressed, and the context forms the setting that the opinion expresses or implies the specific semantic and sentiment towards the target (Liu 2012; Ito et al. 2020). Taking the sentence "*Apple's apps are plentiful*" as an example, the target is "*Apple*", and the context is "*_'s apps are plentiful*" that clarifies the semantic of "*Apple*" being a company and expresses positive sentiment. Therefore, an effective SC model typically needs to make full use of the context information to distinguish the semantics of the target and determine the sentiment polarity of the opinion.

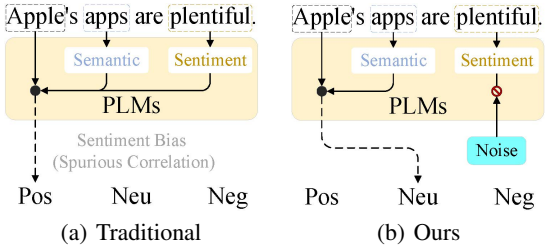
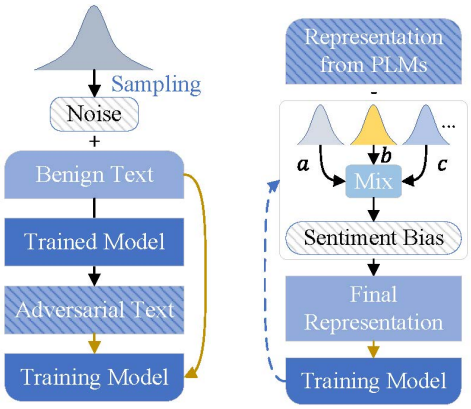


Figure 1: Contextual word embedding of the target (i.e., *Apple*): traditional PLMs *vs.* our method.

With the development of large-scale pre-trained language models (PLMs) such as BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019), ALBERT (Lan et al. 2020) and ELECTRA (Clark et al. 2020), SC models based on fine-tuned PLMs have achieved state-of-the-art performances due to their extraordinary context modeling ability. However, as shown in Figure 1(a), while most PLMs capture the semantic properties of words under diverse contexts (Miaschi and Dell'Orletta 2020; Phan and Ogunbona 2020; Michalopoulos et al. 2022), they also apply the sentiment information of the context to representation learning, thus the contextual word embeddings of some targets are biased toward a specific sentiment. Consequently, spurious correlations between targets and labels are established as shortcuts of decision-making, resulting in the SC model no longer determining the sentiment polarity of the opinion according to the context. Unfortunately, current fine-tuning methods, such as ULMFiT (Howard and Ruder 2018), ITPT (Sun et al. 2019), and IDPT (Sun et al. 2019), mainly focus on performing the cross-domain (or task) knowledge generalization, but neglect to reduce sentiment bias of the target.

Based on the aforementioned, as shown in Figure 1(b), the key point of reducing sentiment bias is to encourage PLMs for encoding the target with the semantic information, without the sentiment information of its context. However, since the self-supervised learning process of PLMs is hard to be controlled, we only try to use adversarial examples to simulate sentiment bias and prompt the classifier to reduce sentiment bias. As shown in Figure 2(a), traditional adversar-

*Corresponding Author.



(a) Traditional (b) Ours

Figure 2: Differences between the computational graphs of traditional and our adversarial attacks/defenses.

ial attacks/defenses keep feeding the benign texts perturbed by noises into the trained classifier and capture adversarial texts, which successfully simulate sentiment bias to attack the trained classifier, to further train new classifier (Xu et al. 2020). However, the following properties of sentiment bias hinder the feasibility of this idea.

- **Complexity:** Every target possibly involves sentiment bias from the other targets. For example, in the sentence "*Apple's excellent products include iPhone and Mac, etc.*", where the entity "*Apple*" may contain sentiment biases from "*iPhone*" and "*Mac*" respectively.
- **Diversity:** Every target possibly involves various sentiment biases. For example, in the sentence "*Apple's products are very expensive*", different from the example in the first paragraph, the sentiment of "*Apple*" is biased towards negative.

To this end, we propose an adaptive **G**umbel-**at**tacked classifi**er** (Gater), as the downstream application of PLMs, to perform debiased sentiment classification in real-time. Firstly, we introduce noises sampled by multiple Gumbel-attack experts to perturb the benign texts. Concretely, each expert maintains a unique Gumbel distribution, which is friendly to the semantic consistency of the benign texts as the type-I generalized extreme value distribution (Lin, Zou, and Ding 2021). Meanwhile, the mixed Gumbel distribution from multiple Gumbel-attack experts can effectively satisfy the complexity of sentiment bias. Secondly, we propose an adaptive trainable framework, which adjusts the number of experts according to the feedback on classification confidence, to satisfy the diversity of sentiment bias. Finally, we leverage a multi-channel parameter updating algorithm to replace the traditional gradient descent algorithm, which effectively integrates the parameter-updating knowledge from multiple experts, to reduce sentiment bias. As shown in Figure 2(b), different from the traditional adversarial attacks/defenses, our method achieves adversarial attacks/defenses end-to-end, without the trained classifier. To summarize, the main contributions of this paper are as follows:

- We propose an adaptive sentiment debiasing method using various Gumbel-attacks, which can flexibly simulate sentiment bias without losing the original semantics of the benign text, and satisfies the diversity and complexity of sentiment bias.
- We propose a multi-channel parameter updating algorithm that can distill the parameter-updating knowledge of multiple expert networks to optimize any traditional gradient descent algorithm.
- Experiments show that our method can be applied to most large-scale pre-trained language models. Unlike other PLMs-based SC methods that require further within-domain or within-task fine-tuning, our method is end-to-end and does not require any extra domain data.

Related Work

Contextual Language Models Contextual representation, output by pre-training language models, has recently led to significant progress in various NLP tasks. In contrast to earlier distributed representation, such as Skip-Gram (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014), contextual representation, such as BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019), ALBERT (Lan et al. 2020) and ELECTRA (Clark et al. 2020), provide diverse embeddings for each word based on different contexts. This enables them to capture the various uses of words across different contexts and encode knowledge that can be transferred across languages. Based on this, numerous effective fine-tuning methods, such as ULMFiT (Howard and Ruder 2018) and IDPT (Sun et al. 2019), have been proposed and have proven successful in enhancing the applicability of contextual representations to downstream tasks.

Bias in Natural Language Processing It is common for machine learning models to inadvertently capture and even amplify unintended dataset biases, which may hurt the generalization of classification models (Qian et al. 2021; Zhao et al. 2017). For example, entity bias often affects false news detection tasks, that is the news pieces containing a certain entity have a strong correlation with the news veracity (Zhu et al. 2022). With the wide application of PLMs, studies on bias of contextual word embeddings have gained significant popularity. The training corpus of PLMs is large-scale, unprocessed real data, containing many social biases such as gender, profession, race, and religion (Nadeem, Bethke, and Reddy 2021). Many studies proved that contextualized word embeddings obtained by PLMs have bias and show how bias propagates to downstream tasks (Bolukbasi et al. 2016; Jentzsch et al. 2019). In the field of sentiment analysis, numerous studies investigated sentiment bias in texts generated by language models, and proposed methods for performing identification and measurement of sentiment bias (Huang et al. 2020). Besides, sentiment bias of each word in the PLMs is also researched by the prompt as a probe, the identification and measurement of bias can improve the effectiveness of fine-tuning methods (Wang et al. 2021; Garg et al. 2022). Different from them, under the premise of ensuring semantic consistency, this paper focuses on reducing sentiment bias of entities that are originally neutral words.

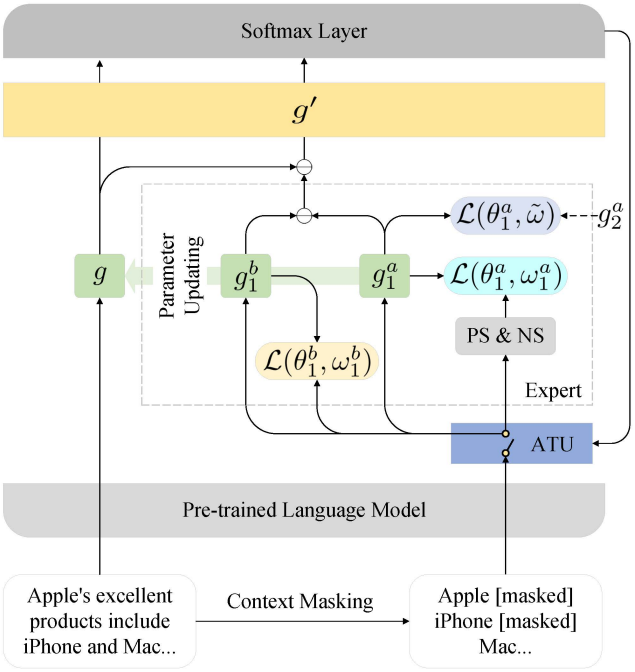


Figure 3: The framework of Gater.

Adversarial Attacks and Defenses The research on adversarial attacks/defenses provides robustness to deep learning algorithms (Sharma et al. 2022). They use various perturbations to analyze the defects existing in the model. Adversarial attacks can be roughly divided into black-box (Swenor 2022; Ye et al. 2022) and white-box attacks (Ebrahimi et al. 2018). In the black-box setting, the adversary tries different perturbations and evaluates the quality of perturbations by querying the model to get the classification result or the output score. In the white-box setting, the adversary has access to the model and thus is capable of generating more sophisticated adversarial examples. Besides, some studies utilized adversarial attacks to detect bias, and proposed the corresponding defense strategies (Emelin, Titov, and Sennrich 2020; Lin, Zou, and Ding 2021). In the field of NLP, the selection of perturbation is based on word-level and character-level transformation. However, the effectiveness of such methods is limited by the uniqueness of entities, that is, the word-level (Swenor 2022) and character-level (Morris et al. 2020) transformations of entities have the potential to drastically change the semantics of entities.

Method

Problem Statement

Let $\mathcal{D} := \{x_m, y_m \mid m = 1, \ldots, M\}$ be a sentiment dataset with M texts x and their labels y. Traditional SC model contains an encoder and a classifier, where the encoder learns a representation $\mathbf{x} \in \mathbb{R}^D$ with the length D for each x, and then the classifier assigns the corresponding prediction label $\hat{y}$ to $\mathbf{x}$. For the SC model based on PLMs, the encoder is replaced by the pre-trained PLMs, and some sentiment biases mislead the classifier to make the error prediction, that is, $\hat{y} \neq y$. In this paper, we prompt the classifier to reduce sentiment bias in the output of the encoder, without modifying its structure.

Overall Architecture

As shown in Figure 3, during the forward propagation phase, Gater adaptively deploys H experts for the original classifier g, and each expert contains an attack-network g_h^a and a base-network g_h^b, where $h \in [1, \ldots, H]$. The inputs of g_h^a and g_h^b are the same text representations with the masked context of entities. g_h^b is responsible for simulating the situation with sentiment bias, while g_h^a is responsible for simulating the situation without sentiment bias.

To train each expert, Gater uses the loss function $\mathcal{L}(\theta_h^b, \omega_h^b)$ to freeze sentiment bias of g_h^b, uses the loss function $\mathcal{L}(\theta_h^a, \omega_h^a)$ to perturb g_h^a with Gumbel noise, and uses loss function $\mathcal{L}(\theta_h^a, \tilde{\omega})$ to increase the diversity of Gumbel noise. Furthermore, each expert obtains the simulated sentiment bias by comparing the outputs of g_h^a and g_h^b, and uses this Gumbel noise to attack the original text representations (i.e., the output of g) without the masked context of entities. Whereafter, the attacked and the original text representations are simultaneously fed into the following full-connected layer g' and softmax layer to calculate their logits respectively. In the beginning of the training epoch, only the first expert (i.e., g_1^a and g_1^b) are activated. Subsequently, Gater uses the adaptive training unit (ATU) to dynamically decide whether to activate the next expert according to the changes in the logits.

During the backward propagation phase, the goal of the multi-channel parameter updating algorithm combines the parameters of g with each g_h^a and g_h^b.

In the test phase, Gater only uses the g and g' to perform the sentiment classification. In other words, experts only participate in the training process of g.

Gumbel-Attack Expert

The goal of the expert is to capture sentiment bias that comes from targets with spurious correlations between them and labels. Therefore, we first perform part-of-speech tagging to mask the context of entities, and then we use Gumbel noise, sampled by g_h^a, to simulate sentiment bias. The obstacle is how to train g_h^a, because we have no training data without sentient bias. To this end, we propose a positive sampling method (PS) and a negative sampling method (NS), respectively. Assuming that the input of both g_h^a and g_h^b is $\mathbf{v}$, and the output of g_h^a is $\mathbf{v}_h^a := \{g_h^a(\mathbf{v}; \theta_h^a)_1, \ldots, g_h^a(\mathbf{v}; \theta_h^a)_D\}$, the $g_h^a(\mathbf{v}; \theta_h^a)_d$ is the value of $\mathbf{v}_h^a$ on the d-th dimension, where $d \in \{1, \ldots, D\}$ and θ_h^a is the parameters of g_h^a. The output of g_h^b is $\mathbf{v}_h^b := \{g_h^b(\mathbf{v}; \theta_h^b)_1, \ldots, g_h^b(\mathbf{v}; \theta_h^b)_D\}$, the $g_h^b(\mathbf{v}; \theta_h^b)_d$ is the value of $\mathbf{v}_h^b$ on the d-th dimension, where $d \in \{1, \ldots, D\}$ and θ_h^b is the parameters of g_h^b. The specific details are as follows:

- **Positive Sampling:** Based on inverse transform sampling (Jang, Gu, and Poole 2017), we first draw noise $\varepsilon := [\varepsilon_1, \ldots, \varepsilon_d]$ from uniform distribution $U(0, 1)$, due

to the cumulative density function of Gumbel distribution is $F(\varrho) = e^{-e^{-\varrho}}$, we can define the Gumbel noise as $\varrho = -\log(-\log(\varepsilon))$, where $\varepsilon \sim U(0,1)$. If we add the Gumbel noise to v_d and derive a new representation $(v_d - \log(-\log(\varepsilon_d)))$, which obeys the Gumbel distribution with location parameter v_d and scale parameter 1. Based on the proof of the Gumbel-Max trick (Jang, Gu, and Poole 2017), Gumbel noise can preserve the semantic consistency of text. That is, our Gumbel-attack reduces sentiment bias without changing the original semantics of neutral words. Based on this, we use softmax function to scale the degree of debiasing, that is,

$$v'_d = \frac{\exp((v_d - \log(-\log(\varepsilon_d)))/\tau)}{\sum_{j=1}^{j=D} \exp((v_d - \log(-\log(\varepsilon_d)))/\tau)}, \quad (1)$$

where $\tau = \frac{h}{10}$. Finally, we perform inverse transform sampling for each v_d in $\mathbf{v}$ to derive the final positive sample $\mathbf{v}_h^+ := \{v'_1, \ldots, v'_D\}$.

- **Negative Sampling:** Negative samples are constructed in the opposite direction to the positive sampling. Therefore, we perform negative sampling in terms of destroying feature permutation importance and improving bias, respectively. Specifically, in each patch of training data, We first randomly sample 50% of text representations to destroy the importance of feature ranking, that is, we shuffle $\mathbf{v}_h^+$. The other 50% of them deteriorate sentiment bias, that is, $v''_d = v_d + \log(-\log(\varepsilon_d))$. Finally, we derive the final negative sample $\mathbf{v}_h^- := \{v''_1, \ldots, v''_D\}$.

For M training texts, we obtain a set of PLMs' outputs $V := \{\mathbf{v}_1, \ldots, \mathbf{v}_M\}$, a set of g_h^a's positive samples $V_h^+ := \{\mathbf{v}_{h,1}^+, \ldots, \mathbf{v}_{h,M}^+\}$, a set of g_h^a's negative samples $V_h^- := \{\mathbf{v}_{h,1}^-, \ldots, \mathbf{v}_{h,M}^-\}$, a set of g_h^a's outputs $V_h^a := \{\mathbf{v}_{h,1}^a, \ldots, \mathbf{v}_{h,M}^a\}$, a set of g_h^b's outputs $V_h^b := \{\mathbf{v}_{h,1}^b, \ldots, \mathbf{v}_{h,M}^b\}$. Then, let V_h^a similar with V_h^+ but different from V_h^-. This goal can be finished with the Mutual Information Neural Estimator (MINE) (Belghazi et al. 2018; Hjelm et al. 2019; Tian et al. 2021; Mroueh et al. 2021). Specifically, we use a discriminator $T_{\omega_h^a}$ with parameters ω_h^a to maximize the mutual information between V_h^a and V_h^+, and the loss function can be defined as follows:

$$\begin{aligned}
\mathcal{L}(\theta_h^a, \omega_h^a) &= \max_{\theta_h^a, \omega_h^a} \mathcal{I}\big(V_h^+, V_h^a\big) \\
&= \mathcal{D}_{KL}\big(p\big(V_h^+, V_h^a\big) \parallel p(V_h^+) \otimes p\big(V_h^a\big)\big) \\
&\geq \hat{\mathcal{I}}^{DV}\big(p(V_h^+); p\big(V_h^a\big)\big) \\
&= \mathbb{E}_{\mathbf{v}_{h,m}^+ \in V_h^+; \mathbf{v}_{h,m}^a \in V_h^a}\Big[T_{\omega_h^a}\big(\mathbf{v}_{h,m}^+; \mathbf{v}_{h,m}^a\big)\Big] \\
&\quad - \log \mathbb{E}_{\mathbf{v}_{h,m}^- \in V_h^-; \mathbf{v}_{h,m}^a \in V_h^a}\Big[e^{T_{\omega_h^a}\left(\mathbf{v}_{h,m}^-; \mathbf{v}_{h,m}^a\right)}\Big].
\end{aligned} \quad (2)$$

We first freeze the parameters θ_h^a of g_h^a and train the discriminator $T_{\omega_h^a}$ based on the Donsker-Varadhan method (DV) (Donsker and Varadhan 1975) to distinguish between samples coming from the joint distribution $p\big(V_h^+, V_h^a\big)$ and the product of marginal distributions $p(V_h^+) \otimes p\big(V_h^a\big)$. Concretely, we train $T_{\omega_h^a}$ as a classifier by using the concatenation between $\mathbf{v}_{h,m}^a$ and $\mathbf{v}_{h,m}^+$ as the positive example, and the concatenation between $\mathbf{v}_{h,m}^a$ and $\mathbf{v}_{h,m}^-$ as the negative example. At a high level, we freeze the parameters ω_h^a of $T_{\omega_h^a}$ to optimize the parameters θ_h^a of g_h^a, maximizing $\mathcal{I}\big(V_h^+, V_h^a\big)$.

Meanwhile, we train g_h^b with the other MINE $T_{\omega_h^b}$ with parameters ω_h^b. The loss function can be defined as follows:

$$\begin{aligned}
\mathcal{L}(\theta_h^b, \omega_h^b) &= \max_{\theta_h^b, \omega_h^b} \mathcal{I}\big(V, V_h^b\big) \\
&= \mathcal{D}_{KL}\big(p\big(V, V_h^b\big) \parallel p(V) \otimes p\big(V_h^b\big)\big) \\
&\geq \hat{\mathcal{I}}^{DV}\big(p(V); p\big(V_h^b\big)\big) \\
&= \mathbb{E}_{\mathbf{v}_m \in V; \mathbf{v}_{h,m}^b \in V_h^b}\Big[T_{\omega_h^b}\big(\mathbf{v}_m; \mathbf{v}_{h,m}^b\big)\Big] \\
&\quad - \log \mathbb{E}_{\tilde{\mathbf{v}}_m \in \tilde{V}; \mathbf{v}_{h,m}^b \in V_h^b}\Big[e^{T_{\omega_h^b}\left(\tilde{\mathbf{v}}_m; \mathbf{v}_{h,m}^b\right)}\Big].
\end{aligned} \quad (3)$$

Similar with the training method of $T_{\omega_h^a}$, we use the concatenation between $\mathbf{v}_m$ and $\mathbf{v}_{h,m}^b$ as the positive example, and the concatenation between $\tilde{\mathbf{v}}_m$ and $\mathbf{v}_{h,m}^b$ as the negative example, where $\tilde{\mathbf{v}}_m$ comes from the shuffled set $\tilde{V}$ of V.

Adaptive Training Architecture

We propose an adaptive training architecture that controls ATU to decide whether to activate more experts based on classification confidence. First, in order to increase the diversity of Gumbel noise, we minimize the mutual information between the noises generated by each expert:

$$\begin{aligned}
\mathcal{L}(\theta_h^a, \tilde{\omega}) &= \min_{\theta_h^a} \max_{\tilde{\omega}} \mathcal{I}\big(V_h^a, V_{ot}\big) \\
&= \mathcal{D}_{KL}\big(p\big(V_h^a, V_{ot}\big) \parallel p(V_h^a) \otimes p(V_{ot})\big) \\
&\geq \hat{\mathcal{I}}_{\tilde{\omega}}\big(p(V_h^a); p\big(V_{ot}\big)\big) \\
&= \mathbb{E}_{\mathbf{v}_h^a \in V_h^a; \mathbf{v}_{ot} \in V_{ot}}\Big[T_{\tilde{\omega}}\big(\mathbf{v}_h^a; \mathbf{v}_{ot}\big)\Big] \\
&\quad - \log \mathbb{E}_{\mathbf{v}_h^a \in V_h^a; \tilde{\mathbf{v}}_{ot} \in \tilde{V}_{ot}}\Big[e^{T_{\tilde{\omega}}\left(\mathbf{v}_h^a; \tilde{\mathbf{v}}_{ot}\right)}\Big],
\end{aligned} \quad (4)$$

where $V_{ot} = \frac{1}{H-1} \sum_{t=1, t \neq h}^{H} V_t^a$, and $\tilde{V}_{ot}$ is the shuffled set of V_{ot}. We set a new MINE $T_{\tilde{\omega}}$ with the concatenation between $\mathbf{v}_h^a$ and $\mathbf{v}_{ot}$ as the positive example, and the concatenation between $\mathbf{v}_h^a$ and $\tilde{\mathbf{v}}_{ot}$ as the negative example. Concretely, we first freeze the parameters θ_h^a of g_h^a, and train the discriminator $T_{\tilde{\omega}}$ to correctly distinguish positive and negative examples. Subsequently, we freeze the parameters $\tilde{\omega}$ of $T_{\tilde{\omega}}$ to optimize g_h^a, minimizing $\mathcal{I}\big(V_h^a, V_{ot}\big)$.

Finally, we use the output of each expert at the t-th training epoch, $V_h^b - V_h^a$, to attack the output V_g of g. Formally, the attacked output can be defined as $V_h = V_g - V_h^b + V_h^a$. Furthermore, V_h and V_g are fed into the following full-connected layer and the softmax layer, and obtain the average confidences $\mathcal{C}_h^t$ and $\mathcal{C}_g^t$. Note that, the confidence is

usually represented by the maximum element of the softmax output, and the average confidence is the mean of the confidence of all samples in classification tasks (Huang et al. 2018; Yang et al. 2020). Based on this, if $\mathcal{C}_g^t \leqslant \lambda$, we add a new expert at the next training epoch. If $\mathcal{C}_g^t > \lambda$, then the expert is no longer added. $\mathcal{C}_h^t$ and $\mathcal{C}_g^t$ are further used during the parameter-updating process.

Multi-Channel Parameter Updating Algorithm

We adapt a novel parameter-updating method about g, which first adapt any traditional gradient-based method as basic parameter-updating method to update the parameters of g and g', then fuse the parameters of each g_h^a and g_h^b into g. For example, we select the traditional stochastic gradient descent algorithm to update the parameters of the g' as follows:

$$W^\top(g')_{t+1} = W^\top(g')_t - \gamma \cdot \nabla_{W^\top(g')_t} \mathcal{L}(W^\top(g')_t), \quad (5)$$

where γ is the learning rate. $W^\top(g')_t$ is the parameters of the network g' at the t-th training-epoch. $\mathcal{L}$ is the selected loss function such as cross-entropy, etc.

Then, for the parameter-updating of g, we first use the gradient descent algorithm to update the parameters of g, and further fuse the parameters of g, each g_h^a and g_h^b. The specific update method is as follows:

$$\begin{aligned} W^\top(g)_{t+1} = {} & W^\top(g)_t - \gamma \cdot \nabla_{W^\top(g)_t} \mathcal{L}(W^\top(g)_t) \\ & + \sum_{h=1}^{H} \mu_h \cdot \gamma \cdot \left(W^\top(g_h^b)_t - W^\top(g_h^a)_t\right), \end{aligned} \quad (6)$$

and

$$\begin{cases} \mu_h = -1, \mathcal{C}_h^t < \mathcal{C}_g^t; \\ \mu_h = 1, \quad \mathcal{C}_h^t > \mathcal{C}_g^t; \\ \mu_h = 0, \quad \mathcal{C}_h^t = \mathcal{C}_g^t. \end{cases} \quad (7)$$

As shown in Equation (7), when $\mu_h = -1$, it indicates that the h-th expert fails to reduce sentiment bias, so the parameter-updating direction of g should be away from that of h-th expert. When $\mu_h = 1$, it indicates that the h-th expert achieves sentiment de-bias, so the parameter-updating direction of g should be close to that of h-th expert. When $\mu_h = 0$, it indicates that the h-th expert has no effect and keeps the original parameter-updating direction of g.

Experiments

Experimental Setups

We connect Gater after PLMs to perform sentiment classification. During the fine-tuning phase, the input of PLMs is a degenerate text-$\varnothing$ pair, where the text comes from each training data and $\varnothing$ means the second sentence is empty. Subsequently, the output of each PLMs is fed into Gater.

Datasets We conducted experiments on seven datasets. IMDb is a binary film review dataset, which is widely used as a benchmark for sentiment classification (Maas et al. 2011). SST-2 is the Stanford Sentiment Treebank (SST) consists of sentences from movie reviews and human annotations of their sentiments (Socher et al. 2013). YELP-2 and YELP-5 are subsets of Yelp's businesses, reviews, and user data, respectively (Xie et al. 2020). Amazon-2 and Amazon-5 are the Amazon review datasets from the Stanford Network Analysis Project, respectively (Xie et al. 2020). SemEval is an English aspect-level sentiment classification, which has 4 pre-defined aspect categories with 4 sentiment polarities (Yang et al. 2021).

Datasets	Train Samples	Test Samples	Classes
IMDb	25,000	25,000	2
SST-2	67,000	1,800	2
YELP-2	560,000	38,000	2
YELP-5	650,000	50,000	5
Amazon-2	3,600,000	400,000	2
Amazon-5	3,000,000	650,000	5
SemEval	6,086	1,600	4

Table 1: The statistics of datasets.

Pre-trained Language Models We selected BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019), ALBERT (Lan et al. 2020) and ELECTRA (Clark et al. 2020) as PLMs. They are all official LARGE versions without any re-pretraining.

Evaluation Metrics On the one hand, we adopt the accuracy metric to evaluate classification performance. On the other hand, we selected the sentiment shift test to detect the sentiment distribution of entities in the PLMs, where the sentiment score of each entity is identified by predicting the sentiment polarity shift after appending it 10 times to various sentiment-oriented reviews (Garg et al. 2022).

Model Settings To ensure a fair comparison, we maintain the same hyper-parameters (e.g., maximum length, warm-up steps, etc.) for each dataset. The only variations made involve tuning the initial learning rate from 1e-5 to 5e-5 for each dataset and adjusting the threshold of the average confidence λ from 0.6 to 0.8. Besides, the number of experts H is set to 7, and the batch size is set to 32. The classifier has a hidden layer of size 50. We use Adam as the basic parameter-updating algorithm with $\beta_1 = 0.9$, $\beta_2 = 0.999$. We tune the number of epochs on the validation set of each dataset. To demonstrate the significance of experimental results, we report the average score and the range of accuracy.

Model Comparisons We selected four fine-tuning methods and four debiasing methods as model comparisons.

- **ULMFiT** proposes multiple fine-tuning techniques to preserve the least general knowledge and avoid catastrophic forgetting during the fine-tuning phase (Howard and Ruder 2018).
- **BERT-ITPT** involves an additional step of pre-training BERT using unsupervised masked language model and next text classification tasks, followed by further fine-tuning of BERT on each dataset (Sun et al. 2019).

Models	IMDb	SST-2	YELP-2	YELP-5	Amazon-2	Amazon-5	SemEval
BERT	95.4	94.9	98.1	70.6	97.3	62.2	84.1
RoBERTa	95.7	96.4	97.9	71.1	96.0	68.1	85.6
ALBERT	94.6	96.9	97.4	67.4	95.3	66.3	84.3
ELECTRA	95.2	96.9	97.7	69.2	95.3	66.8	84.1
ULMFiT	95.4 (—)	–	97.8 (↓ 0.3)	70.0 (↓ 0.6)	–	–	–
BERT+ITPT	95.7 (↑ 0.3)	–	**98.1** (—)	**71.3** (↑ 0.7)	–	–	–
$\text{BERT}_{\text{Pair}}$	–	–	–	–	–	–	**85.9** (↑ 1.8)
UDA	**95.8** (↑ 0.4)	–	97.9 (↓ 0.2)	67.9 (↓ 2.7)	**96.5** (↓ 0.8)	**62.8** (↑ 0.4)	–
Sent-Debias	–	91.5 (↓ 3.4)	–	–	–	–	–
Context-Debias	–	92.8 (↓ 2.1)	–	–	–	–	–
FairFil	–	91.7 (↓ 3.2)	–	–	–	–	–
Auto-Debias	–	**94.0** (↓ 0.9)	–	–	–	–	–
BERT+Gater	95.8 (↑ 0.4)	95.2 (↑ 0.3)	98.1 (—)	71.5 (↑ 0.9)	97.3 (—)	62.8 (↑ 0.6)	86.1 (↑ 2.0)
RoBERTa+Gater	95.7 (—)	96.6 (↑ 0.2)	98.0 (↑ 0.1)	71.1 (—)	96.2 (↑ 0.2)	68.4 (↑ 0.3)	85.8 (↑ 0.2)
ALBERT+Gater	94.7 (↑ 0.1)	97.0 (↑ 0.1)	97.5 (↑ 0.1)	67.5 (↑ 0.1)	95.4 (↑ 0.1)	66.4 (↑ 0.1)	84.5 (↑ 0.2)
ELECTRA+Gater	95.4 (↑ 0.2)	96.9 (—)	97.7 (—)	69.7 (↑ 0.5)	95.3 (—)	66.9 (↑ 0.1)	84.1 (—)

Table 2: Comparison with state-of-the-art results on each dataset. Bold represents the optimal performance in all model comparisons. The underscore indicates that Gater outperforms all systems. ↑ and ↓ represent the performance increase or decrease, respectively. — represents that the performance remains unchanged.

- **$\text{BERT}_{\text{Pair}}$** generates an paired one for each original sentence based on its aspect, and further transforms aspect-based sentiment classification into a task of classifying sentence pairs (Sun, Huang, and Qiu 2019).
- **UDA** improves consistency training by substituting traditional noise injection methods with high-quality data augmentation methods (Xie et al. 2020).
- **Sent-Debias** is a post-processing debias method which aims to remove the estimated gender-direction from sentence representations (Webster et al. 2020).
- **Context-Debias** suggests a debias approach for pre-trained language models (PLM) that involves a loss function designed to promote orthogonality between stereotype words (Kaneko and Bollegala 2021).
- **FairFil** employs a contrastive learning strategy to rectify biases in sentence representations (Cheng et al. 2021).
- **Auto-Debias** presents a modified version of the beam search technique to automatically search for biased prompts (Guo, Yang, and Abbasi 2022).

Main Results

The previous research found that the current debiasing methods might over-debias, leading to downstream tasks' performance degradation (Meade, Poole-Dayan, and Reddy 2022). As shown in Table 2, the accuracy of BERT is reduced from 0.9 to 3.4 by Sent-Debias, Context-Debias, FairFil, and Auto-Debias. In contrast to our method, due to the adaptability of ATU, over-debias can be effectively avoided. As a result, each PLM improved from 0.0 to 2.0 over their original version. Not only that, our approach also outperforms most fine-tuning methods. This shows that sentiment debias for named entities is a feasible fine-tuning method, which makes classifiers rely more on contextual information to make decisions. The training tricks on all datasets are as follows:

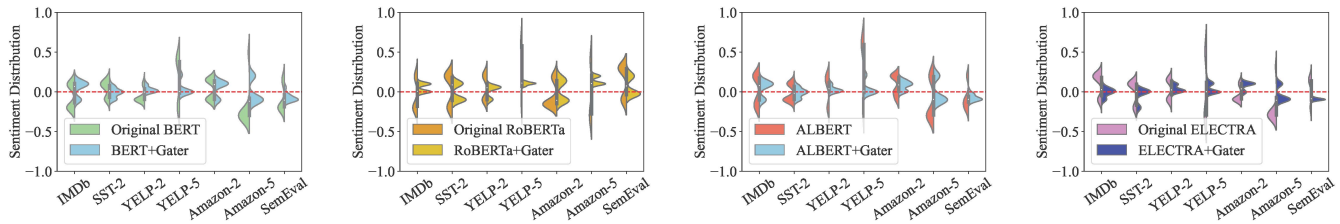
- The threshold of the average confidence can be inspired by the training of the original PLMs. For example, if the average confidence λ of BERT on IMDb is 0.7, the hyper-parameter of BERT+Gater should be set as $0.85 = \lambda + \frac{1-\lambda}{\#\text{Classes}}$.
- We used slanted triangular learning rates (Howard and Ruder 2018), that is, we set different learning rates for each layer of g and g'. Suppose that g and g' have L hidden layers, then we split the parameters W of them into $\{W^1, \ldots, W^L\}$ where W^l contains the parameters of the l-th hidden layer. Then the learning rate γ of Equations (5) and (6) are updated by $\gamma^{l-1} = \xi \cdot \gamma^l$ where the γ^l is the learning rate of W^l, and $\xi \in [0, 1]$ as a decay factor is set by the size of dataset. We found that a larger ξ is more suitable for the small-scale dataset. Concretely, we adopt 0.9, 0.85 and 0.8 for SemEval, IMDb and SST-2, 0.75 and 0.7 for YELP-2 and YELP-5, 0.55 and 0.6 for Amazon-2 and Amazon-5, respectively.

Model Analysis

Sentiment Debiasing We used the NLTK version [1] of the part-of-speech tagging tool to randomly select 400 entities from each dataset, and then calculated the sentiment scores of these entities by sentiment shift testing. The larger the sentiment score, the more severe the sentiment bias.

Figure 4 reflects the change in sentiment bias of these entities before and after using Gater. For each dataset, Gater

[1]NLTK: https://www.nltk.org/



(a) BERT (b) RoBERTa (c) ALBERT (d) ELECTRA

Figure 4: Comparison of sentiment distribution of each language model with and without Gater.

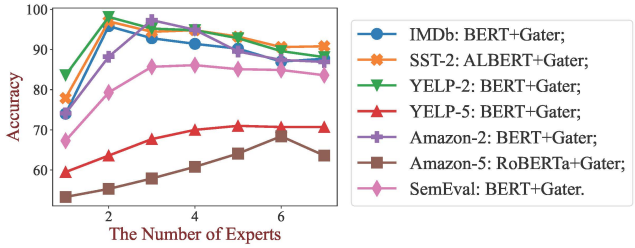
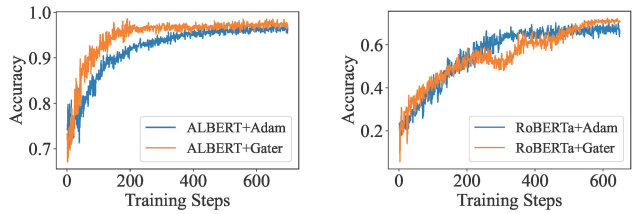


Figure 5: The number of experts on each dataset.



(a) SST-2 (b) Amazon-5

Figure 6: Comparison between parameter updating algorithm and Adam algorithm.

keeps sentiment bias at a low level, that is the sentiment scores of these entities are closer to 0. In contrast, without using Gater, each language model assigned high sentiment scores to these entities. This illustrates Gater effectively reduced sentiment bias. Besides, we found that sentiment bias indeed has a high correlation with accuracy. Concretely, for IMDb, SST-2, YELP-2, and Amazon-2, each language model has slight sentiment bias and obtains high accuracy, and otherwise for the other datasets.

Adaptive Learning Gater adaptively decided the number of experts based on average confidence. Hence, we measured the accuracy of Gater on each dataset when we freeze the adaptive training architecture and manually assign the number of experts.

As shown in Figure 5, the most suitable number of experts is different for each dataset. For IMDb, SST-2, YELP-2, and amazon-2 with slight sentiment bias, Gater only used two or three experts to achieve the best performance. In contrast, for YELP-5 and Amazon-5, Gater needs five or six experts to deal with the severe sentiment bias, respectively. Besides, we found that Gater utilizes up to six experts to outperform or reach the state of the art, and more experts do not achieve higher accuracy. On the contrary, the accuracy drops when we use excessive experts on all datasets. Therefore, it is necessary to adaptively adjust the number of experts.

Parameter Updating We adopt the change rule of accuracy to reflect the parameter updating method when Gater faces different levels of sentiment bias. According to the above analysis, PLMs generally have severe sentiment bias on Amazon-5, and slight sentiment bias on SST-2. Therefore, we evaluate the parameter updating method on these two datasets. For a fair comparison, we adjust the number of parameters of the classifier followed by the original PLMs to be the same as that of Gater. Taking SST-2 as an example, since Gater uses two experts to obtain the best result, the number of parameters of the ALBERT's classifier is equal to the sum of parameters of $g, g_1^a, g_1^b, g_2^a, g_2^b$ and g'. Further, we selected Adam with the same hyper-parameter settings as Gater to update the parameters of ALBERT's classifier.

As shown in Figure 6, for SST-2, the accuracy of Gater and Adam has sharp fluctuations at the initial training steps, but Gater is significantly faster than Adam to achieve the best performance. This illustrates that when sentiment bias is slight, the parameter updating method can effectively shorten the training time. For Amazon-5, compared to Adam, Gater has obvious performance fluctuations from the 250th step to the 400th step, which may be caused by continuously adjusting the number of experts. After the 400th step, the Gater's performance improvement tends to stabilize, and at the 550th step reaches the optimum. This shows that the parameter updating method has the ability to flexibly change the parameter-updating direction and find a better optimal route. Conversely, Adam does not have this flexibility.

Conclusions

In this paper, we propose an adaptive Gumbel-Attack classifier, namely Gater, to reduce sentiment bias in PLMs from an adversarial-attack perspective. As we know, named entities in the real world are complex and diverse, and their features and attributes are subject to change over time and in different circumstances. With the widespread application of PLMs in real-world scenarios, our method enables PLMs to quickly adjust the sentiment orientation of named entities, thereby effectively improving the robustness of sentiment classification systems.

Acknowledgments

This work is supported by the National Key R&D Program of China (2020AAA0108504), the National Natural Science Foundation of China (NSFC) (61972455) and the Joint Project of Bayescom.

References

Belghazi, M. I.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Hjelm, R. D.; and Courville, A. C. 2018. Mutual Information Neural Estimation. In *Proceedings of the 35th International Conference on Machine Learning*, 530–539.

Bolukbasi, T.; Chang, K.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, 4349–4357.

Cheng, P.; Hao, W.; Yuan, S.; Si, S.; and Carin, L. 2021. FairFil: Contrastive Neural Debiasing Method for Pretrained Text Encoders. In *Proceedings of the 9th International Conference on Learning Representations*.

Clark, K.; Luong, M.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the 8th International Conference on Learning Representations*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 22nd Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Donsker, M. D.; and Varadhan, S. S. 1975. Asymptotic Evaluation of Certain Markov Process Expectations for Large Time. *Communications on Pure and Applied Mathematics*, 28(1): 1–47.

Ebrahimi, J.; Rao, A.; Lowd, D.; and Dou, D. 2018. HotFlip: White-box Adversarial Examples for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 31–36.

Emelin, D.; Titov, I.; and Sennrich, R. 2020. Detecting Word Sense Disambiguation Biases in Machine Translation for Model-agnostic Adversarial Attacks. In *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing*, 7635–7653.

Garg, A.; Srivastava, D.; Xu, Z.; and Huang, L. 2022. Identifying and Measuring Token-level Sentiment Bias in Pre-trained Language Models with Prompts. *CoRR*, abs/2204.07289.

Guo, Y.; Yang, Y.; and Abbasi, A. 2022. Auto-Debias: Debiasing Masked Language Models with Automated Biased Prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 1012–1023.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2019. Learning Deep Representations by Mutual Information Estimation and Maximization. In *Proceedings of the 7th International Conference on Learning Representations*.

Howard, J.; and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 328–339.

Huang, G.; Chen, D.; Li, T.; Wu, F.; van der Maaten, L.; and Weinberger, K. Q. 2018. Multi-Scale Dense Networks for Resource Efficient Image Classification. In *Proceedings of the 6th International Conference on Learning Representations*.

Huang, P.; Zhang, H.; Jiang, R.; Stanforth, R.; Welbl, J.; Rae, J.; Maini, V.; Yogatama, D.; and Kohli, P. 2020. Reducing Sentiment Bias in Language Models via Counterfactual Evaluation. In *Findings of the 25th Conference on on Computational Natural Language Learning*, 65–83.

Ito, T.; Tsubouchi, K.; Sakaji, H.; Yamashita, T.; and Izumi, K. 2020. Word-level Contextual Sentiment Analysis with Interpretability. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 4231–4238.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *Proceedings of the 5th International Conference on Learning Representations*.

Jentzsch, S. F.; Schramowski, P.; Rothkopf, C. A.; and Kersting, K. 2019. Semantics Derived Automatically from Language Corpora Contain Human-like Moral Choices. In *Proceedings of the 3rd AAAI/ACM Conference on AI, Ethics, and Society*, 37–44.

Kaneko, M.; and Bollegala, D. 2021. Debiasing Pre-trained Contextualised Embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 1256–1266.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of the 8th International Conference on Learning Representations*.

Lin, J.; Zou, J.; and Ding, N. 2021. Using Adversarial Attacks to Reveal the Statistical Bias in Machine Reading Comprehension Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 333–342.

Liu, B. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Springer.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 142–150.

Meade, N.; Poole-Dayan, E.; and Reddy, S. 2022. An Empirical Survey of the Effectiveness of Debiasing Techniques for Pre-trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 1878–1898.

Miaschi, A.; and Dell'Orletta, F. 2020. Contextual and Non-Contextual Word Embeddings: An In-depth Linguistic Investigation. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, 110–119.

Michalopoulos, G.; McKillop, I.; Wong, A.; and Chen, H. H. 2022. LexSubCon: Integrating Knowledge from Lexical Resources into Contextual Embeddings for Lexical Substitution. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 1226–1236.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 18th Conference on Empirical Methods in Natural Language Processing*, 3111–3119.

Morris, J. X.; Lifland, E.; Yoo, J. Y.; Grigsby, J.; Jin, D.; and Qi, Y. 2020. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. In *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing*, 119–126.

Mroueh, Y.; Melnyk, I.; Dognin, P. L.; Ross, J.; and Sercu, T. 2021. Improved Mutual Information Estimation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 9009–9017.

Nadeem, M.; Bethke, A.; and Reddy, S. 2021. StereoSet: Measuring Stereotypical Bias in Pretrained Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 5356–5371.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing*, 1532–1543.

Phan, M.; and Ogunbona, P. O. 2020. Modelling Context and Syntactical Features for Aspect-based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3211–3220.

Qian, C.; Feng, F.; Wen, L.; Ma, C.; and Xie, P. 2021. Counterfactual Inference for Text Classification Debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 5434–5445.

Sharma, A.; Bian, Y.; Munz, P.; and Narayan, A. 2022. Adversarial Patch Attacks and Defences in Vision-based Tasks: A Survey. *CoRR*, abs/2206.08304.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 18th Conference on Empirical Methods in Natural Language Processing*, 1631–1642.

Sun, C.; Huang, L.; and Qiu, X. 2019. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. In *Proceedings of the 22nd Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 380–385.

Sun, C.; Qiu, X.; Xu, Y.; and Huang, X. 2019. How to Fine-tune BERT for Text Classification? In *Proceedings of the 18th China National Conference*, 194–206.

Swenor, A. 2022. Using Random Perturbations to Mitigate Adversarial Attacks on NLP Models. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 13142–13143.

Tian, J.; Chen, S.; Zhang, X.; Feng, Z.; Xiong, D.; Wu, S.; and Dou, C. 2021. Re-embedding Difficult Samples via Mutual Information Constrained Semantically Oversampling for Imbalanced Text Classification. In *Proceedings of the 26th Conference on Empirical Methods in Natural Language Processing*, 3148–3161.

Wang, B.; Shen, T.; Long, G.; Zhou, T.; and Chang, Y. 2021. Eliminating Sentiment Bias for Aspect-Level Sentiment Classification with Unsupervised Opinion Extraction. In *Findings of the 26th Conference on Empirical Methods in Natural Language Processing*, 3002–3012.

Webster, K.; Wang, X.; Tenney, I.; Beutel, A.; Pitler, E.; Pavlick, E.; Chen, J.; and Petrov, S. 2020. Measuring and Reducing Gendered Correlations in Pre-trained Models. *CoRR*, abs/2010.06032.

Xie, Q.; Dai, Z.; Hovy, E. H.; Luong, T.; and Le, Q. 2020. Unsupervised Data Augmentation for Consistency Training. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*.

Xu, H.; Ma, Y.; Liu, H.; Deb, D.; Liu, H.; Tang, J.; and Jain, A. K. 2020. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. *International Journal of Automation and Computing*, 17(2): 151–178.

Yang, H.; Zeng, B.; Xu, M.; and Wang, T. 2021. Back to Reality: Leveraging Pattern-driven Modeling to Enable Affordable Sentiment Dependency Learning. *CoRR*, abs/2110.08604.

Yang, L.; Han, Y.; Chen, X.; Song, S.; Dai, J.; and Huang, G. 2020. Resolution Adaptive Networks for Efficient Inference. In *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2366–2375.

Ye, M.; Miao, C.; Wang, T.; and Ma, F. 2022. TextHoaxer: Budgeted Hard-label Adversarial Attacks on Text. In *Proceedings of 36th AAAI Conference on Artificial Intelligence*, 3877–3884.

Zhao, J.; Wang, T.; Yatskar, M.; Ordonez, V.; and Chang, K. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In *Proceedings of the 22nd Conference on Empirical Methods in Natural Language Processing*, 2979–2989.

Zhu, Y.; Sheng, Q.; Cao, J.; Li, S.; Wang, D.; and Zhuang, F. 2022. Generalizing to the Future: Mitigating Entity Bias in Fake News Detection. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2120–2125.