Counterfactual Inference for Text Classification Debiasing

Chen Qian
Tsinghua University
qc16@mails.tsinghua.edu.cn

Fuli Feng*
National University of Singapore
fulifeng93@gmail.com

Lijie Wen*
Tsinghua University
wenlj@tsinghua.edu.cn

Chunping Ma
Alibaba DAMO Academy
chunping.mcp@alibaba-inc.com

Pengjun Xie
Alibaba DAMO Academy
chengchen.xpjg@taobao.com

Abstract

Today's text classifiers inevitably suffer from unintended *dataset biases*, especially the document-level *label bias* and word-level *keyword bias*, which may hurt models' generalization. Many previous studies employed data-level manipulations or model-level balancing mechanisms to recover unbiased distributions and thus prevent models from capturing the two types of biases. Unfortunately, they either suffer from the extra cost of data collection/selection/annotation or need an elaborate design of balancing strategies. Different from traditional *factual inference* in which debiasing occurs before or during training, *counterfactual inference* mitigates the influence brought by unintended confounders after training, which can make unbiased decisions with biased observations. Inspired by this, we propose a model-agnostic text classification debiasing framework – CORSAIR, which can effectively avoid employing data manipulations or designing balancing mechanisms. Concretely, CORSAIR first trains a base model on a training set directly, allowing the dataset biases "poison" the trained model. In inference, given a factual input document, CORSAIR imagines its two counterfactual counterparts to distill and mitigate the two biases captured by the poisonous model. Extensive experiments demonstrate CORSAIR's effectiveness, generalizability and fairness. [1]

1 Introduction

Text classification, mapping text documents to a set of predefined categories, is a fundamental and important technique serving for many applications such as sentiment analysis (Qian et al., 2020b), partisanship recognition (Kiesel et al., 2019) and spam detection (Castillo et al., 2007). Machine learning models have become the default choice of solving text classification, owing to their ability to recognize the textual patterns from the labeled documents (Kim, 2014; Howard and Ruder, 2018). Nevertheless, they are at the risk of inadvertently capturing and even amplifying the unintended **dataset biases** (Zhao et al., 2017; Zhang et al., 2020; Feder et al., 2020; Blodgett et al., 2020), which can be at document-level (*i.e.,* **label bias**) and word-level (*i.e.,* **keyword bias**).

The label bias issue occurs in the scenarios where a portion of the categories possesses a majority of training examples than others. For example, the label distribution of a binary sentiment analysis dataset could be 95%:5% (Dixon et al., 2018). Many previous studies found that the models trained on such data are potentially at the risk of simply predicting the majority answers (Dixon et al., 2018; Zhang et al., 2020). The keyword bias issue occurs in the situation where trained models exhibit excessive correlations between certain words and categories, *e.g.,* some sentiment-irrelevant words – "*black*" or "*islam*" – are always connected to negative category. As such, models always lean to unfairly predict any document containing those keywords to a specific category according to the biased statistical information instead of intrinsic textual semantics (Waseem and Hovy, 2016; Liu and Avci, 2019). The serious disadvantages limit models' generalization, especially in the scenarios where the training data is differently-distributed with the testing data (Niu et al., 2021; Goyal et al., 2017).

To resolve the issues, an effective solution is to perform data-level manipulations (*e.g.,* resampling (Qian et al., 2020b)), which effectively transforms a training set to a relatively balanced one before training. Another line of debiasing work typically

*This work was partly done during Chen Qian's internship at Alibaba DAMO academy. Fuli Feng and Lijie Wen are the co-corresponding authors.

[1]The code is available at https://github.com/qianc62/Corsair.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5434–5445
August 1–6, 2021.

designs model-level balancing mechanisms (*e.g.,* reweighting (Zhang et al., 2020)), aiming to adaptively decrease the influence of majority categories while increasing the minority during training. The core of the two types of solutions is to explicitly or implicitly recover unbiased distributions and prevent models from capturing the unintended biases. Unfortunately, the data-level strategy typically suffers from the extra manual cost of data collection, selection and annotation (Zhang et al., 2020), requires much longer training time and normally enlarges the gap between training and testing data distributions. The model-level strategy typically needs elaborate selection or definition of balancing strategies and needs relearning from scratch once certain balancing mechanisms (*e.g.,* an unbiased training objective) are redesigned.

Must machine learning models perform debiasing before or during training? Think about the difference in the decision making processes between machines and humans. Machine learning systems are forced to imitate the behavior from observations via maximizing the prior probability, from which the decision is directly drawn during inference. By contrast, we humans, although born and raised in a biased nature, have the ability of *counterfactual inference* to *make unbiased decisions with biased observations* (Niu et al., 2021). To illustrate, we briefly compare the traditional factual inference and the counterfactual inference in text classification:

• **Factual Inference**: *What will the prediction be if seeing an input document?*

• **Counterfactual Inference**: *What will the prediction be if seeing the main content of an input document only and had not seen the confounding dataset biases?*

The counterfactual inference essentially gifts humans the *imagination ability* (*i.e.,* had not done) to make decisions with a collaboration of the *main content* and the *confounding biases* (Tang et al., 2020), as well as to introspect whether our decision is deceived (Niu et al., 2021), *i.e.,* **counterfactual inference leads to debiased prediction**.

Inspired by this, we propose a novel model-agnostic paradigm (CORSAIR), which adopts factual learning before mitigating the negative influence of the dataset biases in inference (*i.e.,* after training), without the need of employing data manipulations or designing balancing mechanisms. Concretely, in training, CORSAIR directly trains a base model on an original training set, allowing the unintended dataset biases "poison" the model. To "rescue" the testing documents from the poisonous model, in testing, for each factual input document, CORSAIR imagines its two types of counterfactual counterparts to produce two counterfactual outputs as the distilled label bias and keyword bias. Lastly, CORSAIR performs a bias removal operation to produce a counterfactual prediction that corresponds to a debiased decision. To verify, we perform extensive experiments on multiple public benchmark datasets. The results demonstrate our proposed framework's effectiveness, generalizability and fairness, proving that CORSAIR, when employed on four different types of base models, is significantly helpful to mitigate the two types of dataset biases.

2 Methodology

Problem Formalization Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input (text document) and output (category) spaces, respectively. Given a labeled training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}$ (*i.e.,* the observed data), the goal is to learn a text classifier M on $\mathcal{D}_{\text{train}}$, which serves as a mapping function $f(\cdot) : \mathcal{X} \mapsto \mathcal{Y}$ to accurately classify testing examples in $\mathcal{D}_{\text{test}} = \{\hat{x}|\hat{x} \in \mathcal{X}\}$.

Considering that the dataset biases would not be completely eliminated via data manipulations, employing data manipulations (*e.g.,* resampling) or designing balancing mechanisms (*e.g.,* reweighting) may be not a directly-reasonable solution. Inspired by the success of counterfactual inference in mitigating biases in computer vision (Niu et al., 2021; Wang et al., 2020; Tang et al., 2020; Yang et al., 2020; Goyal et al., 2017), we propose a counterfactual-inference-based text-classification debiasing framework (CORSAIR), which is able to make unbiased decisions with biased observations. The core idea of CORSAIR is to train a "poisonous" text classifier regardless the dataset biases and post-adjust the biased predictions according to the causes of the biases in inference. It's worth mentioning that our proposed CORSAIR can be applied to almost any parameterized base model, including traditional one-stage classifiers (*e.g.,* TEXTCNN (Kim, 2014), RCNN (Lai et al., 2015) and LECO (Qian et al., 2020b)) and currently prevalent two-stage classifiers[2] (*e.g.,* ULM-

[2]For brevity, two-stage classifiers refer to two-stage language models with an additional prediction layer.

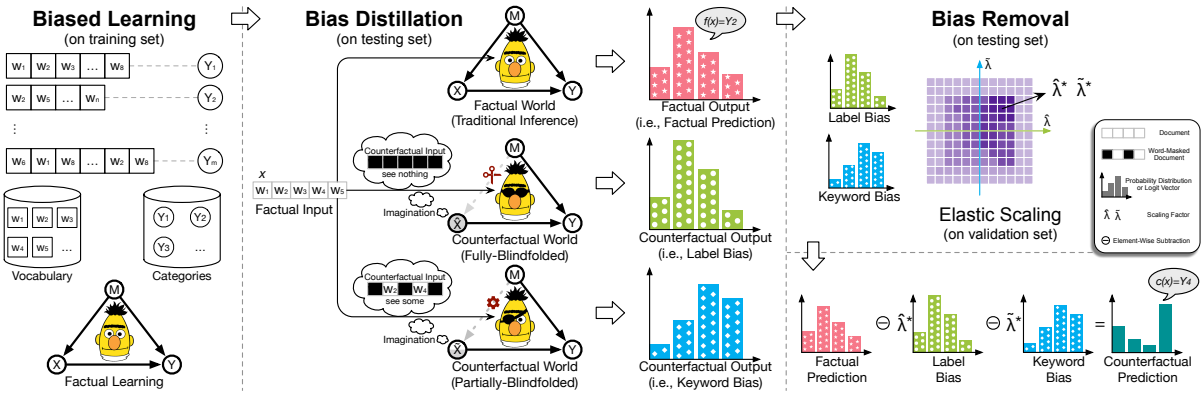

Figure 1: The architecture of our proposed model-agnostic framework (CORSAIR). Specifically, CORSAIR first trains a base model on the training data directly so as to preserve the dataset biases in the trained model. In the inference phase, given a factual input document, CORSAIR first imagines its two types of counterfactual documents to produce two counterfactual outputs as the distilled label bias and keyword bias. Finally, CORSAIR searches two adaptive parameters to perform bias removal to produce a counterfactual prediction for a debiased answer.

FIT (Howard and Ruder, 2018), BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)). For brevity, we will elaborate CORSAIR by taking RoBERTa (a robustly optimized BERT-shape language model) as the example base model, and binary sentiment analysis as the example application. The high-level architecture of CORSAIR is illustrated in Figure 1, which consists of three main components: biased learning, bias distillation and bias removal.

2.1 Biased Learning

In the learning phase (*i.e.,* training), CORSAIR first trains the base model RoBERTa to learn a mapping relation based on training data. Similar to traditional training, CORSAIR uses feedforward to predict batch examples and backward to update those learnable parameters in an end-to-end fashion. In practice, we adopt the standard cross entropy as the training objective (*i.e.,* loss function):

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{y\in\mathcal{Y}} \pi_{i,y} \ln \overline{\pi}_{i,y} \qquad (1)$$
$$\overline{\pi}_i = softmax(f(x_i))$$

where θ denotes the learnable parameters of the base model $f(\cdot)$, n is the number of batch examples, π_i is the ground-truth label distribution (over $\mathcal{Y}$) and $\overline{\pi}_i$ is the predicted probability distribution (over $\mathcal{Y}$) for a given training example x_i.

2.2 Bias Distillation

In the inference phase (*i.e.,* testing), traditional debiasing methods making predictions for each testing document via the conventional feedforward operation on the trained base model to obtain the probability distribution over $\mathcal{Y}$ (*i.e.,* factual prediction) for a most possible answer. However, in addition to the textual contents of the document, the prediction is also affected by unintended *confounders* (Pearl and Mackenzie, 2018) which may produce the label bias and keyword bias. Aiming to obtain unbiased prediction, the key is to debias during inference by *blocking the spread of the biases from learning to inference*. To achieve that, inspired by the counterfactual studies in causal reasoning (Niu et al., 2021; Tang et al., 2020), we design an effective strategy based on *causal intervention* (Pearl, 2013; Pearl and Mackenzie, 2018) to distill the potentially-harmful biases captured by the trained model (Niu et al., 2021; Tang et al., 2020), and then mitigate them via bias removal.

2.2.1 Causal Graph

Aiming to conduct proper causal intervention, we first formulate the *causal graph* (Pearl, 2013; Pearl and Mackenzie, 2018; Tang et al., 2020) for the text classification models (see the left-bottom part of Figure 1), which sheds light on how the document contents and dataset biases affecting the prediction. Formally, a causal graph is a directed acyclic graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, indicating how a set of variables $\mathcal{N}$ causally interact with each other through the causal links $\mathcal{E}$. It provides a sketch of the causal relations behind the data and how variables obtain their values (Tang et al., 2020), *e.g.,* $(X, M) \rightarrow Y$. In this causal graph, X, Y and M denote a text document's embedding, its corresponding prediction and the trained model which

inevitably captures unintended confounders existing in training data, respectively.

2.2.2 Label Bias Distillation

According to the causal graph, we diagnose how the dataset biases existing in training data misleads inference. Concretely, by using Bayes rule (Wang et al., 2020), we can view the inference as:

$$f(x) = P(Y|X) = \sum_{c} P(Y|X,c)P(c|X) \quad (2)$$

where c could be any confounder captured by the model trained on a biased training set (*e.g.,* the overwhelming majority of training documents fall in POSITIVE). Under such circumstances, once the training documents corresponding to the POSITIVE category are dominating than NEGATIVE, the trained model tends to build strong spurious connections between testing documents and POSITIVE, achieving high accuracy even without knowing testing documents' main contents. As such, the model is inadvertently contaminated by the spurious causal correlation: $X \leftarrow M \rightarrow Y$, *a.k.a.* a *back-door path* in causal theory (Pearl and Mackenzie, 2018; Pearl, 2013). To decouple the spurious causal correlation, the *back-door adjustment* (Pearl and Mackenzie, 2018; Pearl, 2013; Pearl et al., 2016) predicts an actively intervened answer via the $do(\cdot)$ operation:

$$P(Y|do(X)) = P(Y|X = \hat{x}) = f(\hat{x}) \quad (3)$$

where $\hat{x}$ could be any counterfactual embedding as long as it is no longer dependent on M to detach the connection between X and M. As illustrated in the fully-blindfolded counterfactual world in Figure 1, the causal intervention operation wipes out all the in-coming links of a cause variable X, which encourages the model M to inference without seeing any testing document, *i.e.,* RoBERTa should be fully blind in order to detaching the connection between M and X. To achieve that, we use $\hat{x}$ to denote the imagined fully-blindfolded counterfactual document where all words in the test document x are consistently masked (to create a counterfactual embedding), and $f(\hat{x})$ as the corresponding counterfactual output via feedforward through the trained model. Since the model cannot see any word in the factual input x after fully blindfolding, $f(\hat{x})$ actually reflects the pure influence from the trained base model M. Furthermore, $f(\hat{x})$ refers to the output (*e.g.,* a probability distribution or a logit vector) where no textual information is given. Thus, the fully-blindfolded counterfactual output:

$$P(Y|do(X)) = f(\hat{x}) = f(\langle w_1, w_2, \cdots, w_n \rangle) \quad \forall w_i \in \hat{x}, w_i \leftarrow \texttt{[MASK]} \quad (4)$$

naturally reflects as the label bias captured by M, where `[MASK]` is a special token to mask a single word. Due to $\hat{x}$ is fully-blindfolded and independent with trained model M, in implementation, we follow Wang et al. (2020) to use the average document feature on the whole training set as its embedding of the counterfactual document.

2.2.3 Keyword Bias Distillation

Inspired by the factual inference where all textual information in test documents are exposed to the base model and the fully-blindfolded case where all textual information in each test document are not exposed, we make the first attempt to utilize a partially-blindfolded counterfactual document where some words in the test document x are masked to distill the keyword bias from the trained base model.

Specifically, we deliberately expose some words which may potentially cause spurious correlations (*e.g.,* the spurious "*black*"-to-NEGATIVE mapping) to the trained model to exhibit their potentially negative influence. Some evil words may serve as unintended confounders (Tang et al., 2020), splitting a document into two pieces: main *content* and relatively-unimportant *context*. In the following, we use $\tilde{x}$ to denote another counterfactual document where the main-content words in a test document x are masked while other context words are not, and $f(\tilde{x})$ as the corresponding counterfactual output. To achieve that, an effective masking strategy is to use discriminative *text summarization* methods to extract the main content of the document, before masking content words (important classification clues) and exposing others as potentially harmful biasing factors. Since the model is forced to see only the non-masked context words in x, $f(\tilde{x})$ actually reflects the influence from both the potentially harmful contexts and the trained model. Thus, the partially-blindfolded counterfactual output:

$$f(\tilde{x}) = f(\langle w_1, w_2, \cdots, w_n \rangle) \quad \forall w_i \in \tilde{x}, \begin{cases} w_i \leftarrow \texttt{[MASK]} & \text{if } w_i \in x_{\text{content}} \\ w_i \leftarrow w_i & \text{if } w_i \in x_{\text{context}} \end{cases} \quad (5)$$

naturally reflects as the keyword bias captured by M for a specific text document x, where x_{content} and x_{context} denote the main content and the con-

text of x, respectively. Inspired by a recent counterfactual word-embedding study of Feder et al. (2020), to realize discriminative text summarization, we use Jieba[3] tool, whose TextRank-based interface can effectively extract the words that may influence the semantics of a sentence as content, leaving potentially discriminative/unfair keywords (*e.g.,* stop words, a part of adjectives, and semantically-unimportant particles) as contexts. Empirically, the average ratio of contents to contexts produced by Jieba on all datasets is approximately 62.03%:37.97%.

2.3 Bias Removal

Our final goal is to use the direct effect from X to Y for debiased prediction, removing ($\backslash$) the label bias and the keyword bias existing in training data (*i.e.,* blocking the spread of the biases from training data to inference): $f(x)\backslash f(\hat{x})\backslash f(\tilde{x})$. The debiased prediction via bias removal can be formalized via the conceptually simple and empirically powerful element-wise subtraction operation:

$$c(x) = f(x)\backslash f(\hat{x})\backslash f(\tilde{x}) = f(x) - \hat{\lambda} f(\hat{x}) - \tilde{\lambda} f(\tilde{x}) \quad (6)$$

where $f(x)$ and $c(x)$ correspond to the traditional factual prediction and our counterfactual prediction, respectively; $f(\hat{x})$ and $f(\tilde{x})$ correspond to the label bias and the keyword bias distilled from the trained base model, respectively; $\hat{\lambda}$ and $\tilde{\lambda}$ are two independent parameters balancing the two types of biases.

Note that the two distilled biases could be probability distributions over all categories or logit vectors (*i.e.,* without normalization), and they typically do not contribute completely equally to the final classification. As such, in Equation 6, directly subtracting without adaptive parameters (*i.e.,* $\hat{\lambda}$=$\tilde{\lambda}$=$\frac{1}{2}$) would cause that mitigating a certain bias too much or too less for a specific testing set. Therefore, we propose the *elastic scaling* mechanism to search two adaptive parameters (*scaling factors*) – $\hat{\lambda}^*$ and $\tilde{\lambda}^*$ – on the *validation set* to amplify or penalize the two biases, which would dynamically adapt to different datasets according to the extent to which two biases in training set "poison" the validation set. In practice, elastic scaling can be implemented using *grid beam search* (Hokamp and Liu, 2017) in a scoped two-dimensional space:

$$\hat{\lambda}^*, \tilde{\lambda}^* = \arg\max_{\hat{\lambda},\tilde{\lambda}} \psi(\mathcal{D}_{\text{dev}}, c(x; \hat{\lambda}, \tilde{\lambda})) \quad \hat{\lambda}, \tilde{\lambda} \in [a, b] \quad (7)$$

where ψ is a metric function (*e.g.,* recall, precision and F_1-score) to evaluate the performance on the validation set $\mathcal{D}_{\text{dev}}$=$(X_{\text{dev}}, Y_{\text{dev}})$; a and b are the boundaries of the search range. The two factors are at dataset-level and thus searched only once for each validation set, and would be used in inference for all testing documents.

3 Evaluation

Baselines We choose four types of representative text classifiers as the base models of our proposed framework, covering classical, data-manipulation-based, model-balancing-based, as well as large-scale and two-stage methods. TEXTCNN (Kim, 2014) is a classical classifier that uses convolutional neural networks (CNN) with scale-variant convolution filters to capture local textual features, which may potentially capture spurious correlations between certain keywords and categories. LECO (Qian et al., 2020b) utilizes the combination of the implicit encoding of deep linguistic information and the explicit encoding of morphological features, which would also capture the keyword bias inadvertently. Besides, it uses a sentence-level over-sampling mechanism (He and Garcia, 2009) to mitigate the label bias, and we further enhance it via a powerful word-level augmentation technique (EDA) (Wei and Zou, 2019) to mitigate the keyword bias, denoted as LECOEDA. WEIGHT (Zhang et al., 2020) is a most recent debiasing text classifier that uses a specially-designed reweighting technique under an unbiased objective for fair (*i.e.,* non-discrimination) learning, which is proven effective to mitigate the unfairness or discrimination issue caused by unintended dataset biases. RoBERTa (Liu et al., 2019) is an improved version of BERT, whose effective modifications allow RoBERTa to generalize better and match or exceed the performance of many post-BERT methods, serving as a very strong baseline in recent work (Gururangan et al., 2020).

Datasets We use multiple English benchmark datasets (used mainly in academic community): HyperPartisan (Kiesel et al., 2019), Twitter (Huang et al., 2017), ARC (Jurgens et al., 2018), SCIERC (Luan et al., 2018), ChemProt (Kringelum et al., 2016), Economy (Huang and Paul, 2018), News (Lang, 1995), Parties (Huang and Paul, 2018), YelpHotel (Zhang et al., 2014); and also randomly collect real-world query-

[3] https://github.com/fxsjy/jieba

Table 1: Statistics of the datasets. #D denotes the average number of characters per document. #C denotes the number of categories. #Train, #Dev and #Test denote the number of training set, validation set and testing set, respectively.

Dataset	Domain/Genre	#D	#C	#Train↑	#Dev	#Test
HYP	Political News	3,265.64	2	516	64	65
TWI	Social Network	84.32	2	1,631	272	272
ARC	Computer Science	222.49	6	1,688	125	128
SCI	Computer Science	192.92	7	3,219	712	717
CHE	Biomedicine	220.28	13	4,169	2,944	2,952
ECO	Finance	1,152.22	2	4,744	595	596
NEW	News	1,801.20	20	9,445	4,689	4,694
PAR	Political Speech	140.31	2	10,059	2,012	2,012
YEL	User Comment	651.73	3	20,975	6,991	6,993
TAO	E-Commerce	8.09	143	68,086	6,949	7,022
SUN	E-Commerce	7.70	56	234,074	50,851	50,844

category pairs (used in industrial community) from two famous Chinese e-commerce platforms: Taobao[4] and Suning[5]. For brevity, we will use the first three letters to denote each dataset (*e.g.,* HYP for HyperPartisan). The statistics of the datasets are summarized in Table 1.

Metric We use the widely-used *macro-F_1* metric, which is the balanced harmonic mean of *precision* and *recall*. Furthermore, macro-F_1 is more suitable than *micro-F_1* to reflect the extent of the dataset biases, especially for the highly-skewed cases, since macro-F_1 is strongly influenced by the performance in each category (*i.e.,* category-sensitive) but micro-F_1 easily gives equal weight over all documents (*i.e.,* category-agnostic) (Kim et al., 2019).

Implementation Details The search range in Equation 7 is set as $[-2.0, 2.0]$. Each training is run for 10 epochs with the Adam optimizer (Kingma and Ba, 2015), a mini-batch size of 16, a learning rate of $2e^{-5}$, and a dropout rate of 0.1. We implement CORSAIR via Python 3.7.3 and Pytorch 1.0.1. All of our experiments are run on a machine equipped with seven standard NVIDIA TITAN-RTX GPUs.

3.1 Overall Performance

We report the average results over five different initiations in Table 2. We can observe that CORSAIR consistently improves the four types of representative baselines on almost all datasets with a significance level, regardless of the languages, domains, volumes and applications of the datasets, which validates the effectiveness and the generalizability of the proposed framework. Furthermore, since CORSAIR performs debiasing between the traditional factual predictions and two counterfactual outputs to produce counterfactual predictions, the comparison between each baseline and its CORSAIR-equipped counterparts highlights the importance of the counterfactual inference, which is largely ignored by most of previous text classification methods. Particularly, CORSAIR can even benefit the data-manipulation-based method (*i.e.,* LECOEDA) and the model-balancing-based method (*i.e.,* WEIGHT) consistently, which in turn verifies our initial intuition that the dataset biases would not be completely eliminated via data manipulations merely, and further illuminates our key insight – preserving biases in models before debiasing in inference.

We can also notice that CORSAIR sometimes hurts performance (*e.g.,* RoBERTa+CORSAIR on HYP and ARC); we conjecture the phenomenon comes from the small-scale data, making the giant model RoBERTa overfits and thus "fail" to distill two potential biases that are identically distributed with the ideal distributions of factual biases. Moreover, finetuning a RoBERTa model on large-datasets (*e.g.,* SUN) would take about 36 hours, nearly 50 times that of training a WEIGHT model (about 44 minutes); we thus suggest to use lightweight base models in practice with considering systems' robustness and efficiency. Besides, the proposed framework works only in inference and can thus be employed on the previous already-trained models. Therefore, by leveraging counterfactual inference, our approach can serve as a powerful, "data-manipulation-free" and "model-balancing-free" weapon to enhance different types of text classification methods.

3.2 Bias Analysis

According to Sweeney and Najafian (2020), the more imbalanced/skewed a prediction produced by a trained model is, the more unfair opportunities it gives over predefined categories, the more unfairly-discriminative the trained model is. We thus follow previous work (Xiang and Ding, 2020; Sweeney and Najafian, 2020) to use the metric – *imbalance divergence* – to evaluate whether a prediction (normally a probability distribution) P is imbalanced/skewed/unfair:

$$D(P, U) = JS(P||U) \tag{8}$$

where $D(\cdot)$ is defined as the distance of P and the uniform distribution U (with $|P|$ elements). Concretely, we use the JS divergence as the distance

[4] https://www.taobao.com
[5] https://www.suning.com

Table 2: Experimental results (F_1; %) of all methods on all benchmark datasets (higher is better). For each dataset, the best-performing results among all methods are highlighted with boldfaces. For each baseline, the best-performing results between the baseline and our approach are highlighted with $*$. † denotes statistical significance ($p \leq 0.05$) between a baseline and the counterpart employed on our framework.

Method	HYP	TWI	ARC	SCI	CHE	ECO	NEW	PAR	YEL	TAO	SUN	AVG.	Δ
TEXTCNN	40.48	65.94	12.46	10.09	18.96	46.07	12.07	54.94	51.49	08.16	10.90	30.14	–
TEXTCNN+CORSAIR	46.71$^{\dagger}_{*}$	69.03$^{\dagger}_{*}$	17.03$^{\dagger}_{*}$	19.85$^{\dagger}_{*}$	22.55$^{\dagger}_{*}$	59.74$^{\dagger}_{*}$	16.18$^{\dagger}_{*}$	56.39$^{\dagger}_{*}$	58.37$^{\dagger}_{*}$	08.70$_{*}$	14.20$^{\dagger}_{*}$	35.34$^{\dagger}_{*}$	5.20↑
LECOEDA	58.78	72.43	52.64	22.37	30.22	60.81	54.39	57.33	60.60	12.02	17.17	45.34	–
LECOEDA+CORSAIR	60.46$^{\dagger}_{*}$	74.62$^{\dagger}_{*}$	53.10$^{\dagger}_{*}$	23.28$_{*}$	30.42$_{*}$	**61.81**$_{*}$	54.48$_{*}$	57.51$_{*}$	60.87$_{*}$	14.25$^{\dagger}_{*}$	22.62$^{\dagger}_{*}$	46.67$^{\dagger}_{*}$	1.33↑
WEIGHT	49.14	60.80	12.71	09.80	11.98	44.67	15.19	54.90	45.73	01.67	06.54	28.46	–
WEIGHT+CORSAIR	55.03$^{\dagger}_{*}$	68.35$^{\dagger}_{*}$	18.04$^{\dagger}_{*}$	17.73$^{\dagger}_{*}$	22.08$^{\dagger}_{*}$	59.24$^{\dagger}_{*}$	20.93$^{\dagger}_{*}$	55.70$_{*}$	58.47$^{\dagger}_{*}$	06.54$^{\dagger}_{*}$	14.02$^{\dagger}_{*}$	36.01$^{\dagger}_{*}$	7.55↑
RoBERTa	**87.92**$_{*}$	88.71	**68.76**$_{*}$	81.76	50.10	53.55	85.38	65.54	77.67	50.70	44.05	68.55	–
RoBERTa+CORSAIR	86.45	**89.12**$_{*}$	68.10	**82.21**$_{*}$	**51.65**$_{*}$	61.31$^{\dagger}_{*}$	**86.83**$^{\dagger}_{*}$	**67.09**$^{\dagger}_{*}$	**77.69**$_{*}$	**51.52**$^{\dagger}_{*}$	**46.15**$^{\dagger}_{*}$	**69.82**$^{\dagger}_{*}$	1.27↑

Table 3: Experimental results (imbalance divergence or unfairness; %) of all methods on all benchmark datasets (lower is better). The top subtable shows the average document-level imbalance of predictions for label bias evaluation, and the bottom one shows the average word-level imbalance of predictions for keyword bias evaluation.

	Method	HYP	TWI	ARC	SCI	CHE	ECO	NEW	PAR	YEL	TAO	SUN	AVG.	Δ
Label Imbalance (RLI)	TEXTCNN	01.39	06.31	11.88	09.99	18.86	06.62	28.21$_{*}$	01.41$_{*}$	09.43	41.87$_{*}$	46.12$_{*}$	16.55	–
	TEXTCNN+CORSAIR	01.07$_{*}$	05.18$^{\dagger}_{*}$	02.27$^{\dagger}_{*}$	01.62$^{\dagger}_{*}$	11.53$^{\dagger}_{*}$	01.52$^{\dagger}_{*}$	28.49	01.49	09.23$_{*}$	42.01	46.77	13.74$^{\dagger}_{*}$	2.81↓
	LECOEDA	01.11$_{*}$	07.47$^{\dagger}_{*}$	10.42$^{\dagger}_{*}$	11.08$_{*}$	08.93$_{*}$	03.51$_{*}$	05.36$^{\dagger}_{*}$	00.64$_{*}$	06.66	26.91	22.25$_{*}$	09.48$^{\dagger}_{*}$	–
	LECOEDA+CORSAIR	01.21	11.29	12.96	11.99	09.26	04.47	06.05	00.72	05.08$_{*}$	26.06$^{\dagger}_{*}$	23.05	10.19	0.71↑
	WEIGHT	00.81$_{*}$	03.19	07.06	05.10	12.65	03.81	01.99	00.18	02.43	25.71	34.76	08.88	–
	WEIGHT+CORSAIR	00.88	01.66$^{\dagger}_{*}$	01.95$^{\dagger}_{*}$	00.98$^{\dagger}_{*}$	04.68$^{\dagger}_{*}$	00.56$^{\dagger}_{*}$	01.30$^{\dagger}_{*}$	00.16$_{*}$	01.21$^{\dagger}_{*}$	14.08$^{\dagger}_{*}$	14.01$^{\dagger}_{*}$	03.77$^{\dagger}_{*}$	5.11↓
	RoBERTa	01.29	02.96	14.57	18.10	16.74	06.69	00.16	00.01$_{*}$	02.55	57.74	56.76	16.14	–
	RoBERTa+CORSAIR	00.11$^{\dagger}_{*}$	01.27$^{\dagger}_{*}$	01.66$^{\dagger}_{*}$	12.57$^{\dagger}_{*}$	02.76$^{\dagger}_{*}$	02.15$^{\dagger}_{*}$	00.02$_{*}$	00.01$_{*}$	00.82$^{\dagger}_{*}$	28.83$^{\dagger}_{*}$	22.91$^{\dagger}_{*}$	06.64$^{\dagger}_{*}$	9.50↓
Keyword Imbalance (RKI)	TEXTCNN	17.96	17.39	44.76	47.39	37.35	20.69	38.23	05.76	18.46	65.37	60.87	34.02	–
	TEXTCNN+CORSAIR	07.44$^{\dagger}_{*}$	15.17$^{\dagger}_{*}$	29.36$^{\dagger}_{*}$	22.36$^{\dagger}_{*}$	28.84$^{\dagger}_{*}$	08.51$^{\dagger}_{*}$	35.80$^{\dagger}_{*}$	05.09$_{*}$	12.02$^{\dagger}_{*}$	64.81$^{\dagger}_{*}$	58.37$^{\dagger}_{*}$	26.16$^{\dagger}_{*}$	7.86↓
	LECOEDA	06.77	11.93$^{\dagger}_{*}$	26.54	15.01	24.16	07.71	30.05	05.09	12.39$_{*}$	65.30	60.63	24.14	–
	LECOEDA+CORSAIR	06.61$_{*}$	14.46	25.94$_{*}$	14.13$^{\dagger}_{*}$	22.53$^{\dagger}_{*}$	04.77$^{\dagger}_{*}$	30.03$_{*}$	05.05$_{*}$	12.58	57.51$^{\dagger}_{*}$	52.98$^{\dagger}_{*}$	22.41$^{\dagger}_{*}$	1.73↓
	WEIGHT	10.32	18.77	43.64	47.70	46.53	21.29	38.98	06.30	21.34	66.75	61.73	34.85	–
	WEIGHT+CORSAIR	06.34$^{\dagger}_{*}$	13.70$^{\dagger}_{*}$	33.29$^{\dagger}_{*}$	23.40$^{\dagger}_{*}$	28.97$^{\dagger}_{*}$	08.80$^{\dagger}_{*}$	34.74$^{\dagger}_{*}$	05.32$_{*}$	10.12$^{\dagger}_{*}$	64.87$^{\dagger}_{*}$	58.63$^{\dagger}_{*}$	26.19$^{\dagger}_{*}$	8.66↓
	RoBERTa	21.58	21.58	45.39	41.57	54.57	21.58	59.26	21.58	31.83	67.23	64.82	40.99	–
	RoBERTa+CORSAIR	19.40$^{\dagger}_{*}$	13.52$^{\dagger}_{*}$	35.87$^{\dagger}_{*}$	34.19$^{\dagger}_{*}$	53.37$_{*}$	18.99$^{\dagger}_{*}$	55.82$^{\dagger}_{*}$	17.74$^{\dagger}_{*}$	30.52$_{*}$	62.23$^{\dagger}_{*}$	60.82$^{\dagger}_{*}$	36.58$^{\dagger}_{*}$	4.41↓

metric since it is symmetric (*i.e.*, $JS(P||U) = JS(U||P)$) and strictly scoped (in $[0.0, 1.0]$) compared with the KL divergence. Based on this, to evaluate the label bias and the keyword bias of a trained model M, we average its relative *label imbalance* (RLI) over the predicted distributions of all the testing documents, and the relative *keyword imbalance* (RKI) over all the testing documents containing whichever context word, respectively:

$$RLI(M) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} D(P(x), U)$$
$$RKI(M, \mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{w \in \mathcal{V}} D(P(\{x | w \in x \wedge x \in \mathcal{D}\}), U) \quad (9)$$

where a prediction $P(x)$ could be a factual prediction $f(x)$ or a counterfactual one $c(x)$; $\mathcal{V}$ denotes the vocabulary of context words. The two metrics implicitly capture the distance between all predictions and the fair uniform distribution U.

Table 3 shows the average results of the bias analysis investigation over five different initiations. The results show that our framework reduces the imbalance metrics (lower is better) when employed on non-data-balanced baselines significantly and consistently, indicating it is indeed helpful to mitigate the two dataset bias issues. We all know that data-balanced LECOEDA perfectly mitigates the label bias issue via data balancing, thus achieving the lowest RLI. Due to the powerful debiasing operations via strictly balancing data, it serves as the skyline of RLI. This finding is similar to previous evidence of Morik et al. (2020). Moreover, we can also see that LECOEDA reduces the RKI, validating that data manipulation methodology is indeed helpful to debias the keyword bias issue but fails to eliminate it completely; our framework can further reduce RKI (1.73↓). Note that WEIGHT exhibits a more severe keyword bias than label bias (34.85 vs. 08.88). The key reason is that WEIGHT explicitly balances each category according to a theoretically fair objective but ignores the consideration of label distributions conditioned on finer-grained words. Moreover, RoBERTa exhibits the

most imbalanced prediction against all baselines and across small- and large-scale datasets (*e.g.,* ARC and TAO), indicating that its answers excessively distribute on certain categories due to the overfitting phenomenon rooted from its large-scale parameters (about 110M). Luckily, by being equipped with our framework, the RoBERTa case remarkably reduces the imbalance issue caused by dataset biases (9.50↓ and 4.41↓).

Another finding is that the keyword bias issue typically is more severe than the label bias, meaning that trained models typically utilize the word-level information to inference, which could catch angel keywords as good clues but also inevitably utilize evil keywords that are potential biases. Additionally, the keyword bias issue, compared with label bias, is much harder to be completely eliminated via data manipulations, which imposes a caution for relevant studies to keep a watchful eye on the detrimental causal correlations.

3.3 Ablation Study

We conduct ablation studies on CORSAIR to empirically examine the contribution of its main mechanisms/components, including the label bias removal operation (\LBR), the keyword bias removal operation (\KBR) and the elastic scaling mechanism (\ES).

The average results of the ablation study are shown in Table 4. We can see that removing the proposed CORSAIR causes serious performance degradation, dropping F_1-score by 7.55 points for the WEIGHT case. Additionally, it also provides evidence that using the counterfactual framework for text classification can explicitly mitigate two types of dataset biases to generalize better on unseen examples. Moreover, we observe that mitigating the two types of biases are consistently helpful for classification tasks. The key reason is that the distilled label bias provides a global (*i.e.,* document-agnostic) offset and the distilled keyword bias provides a local (*i.e.,* document-specific) one to "move" in the predicted space, which makes the trained models "blind" to see potentially harmful biases existing in observed data so as to focus only on the main content of each document to inference. Meanwhile, elastic scaling effectively finds two dynamic scaling factors to amplify or shrink two biases, making the biases be mitigated properly and adaptively.

Table 4: Ablation study on main components or mechanisms of our framework evaluated on all datasets. \ denotes the removing operation. ↓ denotes performance drop. The worst scores are underlined.

LECOEDA+CORSAIR	46.67	Δ	WEIGHT+CORSAIR	36.01	Δ
\CORSAIR	45.34†	1.33↓	\CORSAIR	<u>28.46</u>†	<u>7.55</u>↓
\LBR	<u>40.82</u>†	<u>5.85</u>↓	\LBR	33.05†	2.96↓
\KBR	45.30†	1.37↓	\KBR	30.05†	5.96↓
\ES	43.97†	2.70↓	\ES	32.85†	3.16↓

3.4 Further Investigation on Counterfactual Learning

Recall that our proposed framework first trains a base model on a training set directly (factual learning) so as to preserve dataset biases in the trained model, and in the inference phase, given a factual input document, CORSAIR imagines two types of counterfactual documents aiming to produce two counterfactual outputs as the distilled label bias and keyword bias for bias removal. That is, the framework deliberately causes the discrepancy between learning and inference, leading to an operational gap between the two phases. In this section, we investigate more deeply to explore what will happen if the operational gap is bridged.

• **Factual Learning**. Learn with $\mathcal{L}(\theta; f(x_i), y_i)$ as objective, *i.e.,* to minimize the loss between factual predictions and ground-truth labels. Then, inference via counterfactual predictions.

• **Counterfactual Learning**. Learn with $\mathcal{L}(\theta; c(x_i), y_i)$ as objective, *i.e.,* to minimize the loss between counterfactual predictions and ground-truth labels. Then, inference directly.

The average results of TEXTCNN on ECO ($|\mathcal{Y}|$=2) and CHE ($|\mathcal{Y}|$=13) are reported in Figure 2. We observe that these configurations converge at different F_1 scores as the number of epochs increases gradually. As for each dataset, the configuration of a factual model with counterfactual inference (*i.e.,* CORSAIR) achieves the best performance with even a relatively more rapid convergence. More interestingly, in the early phases of model training (*e.g.,* epoch=0), CORSAIR usually provides a higher starting point than traditional factual inference. We conjecture that the superiority may come from the use of average embedding which usually produces a stable distribution similarly distributed with ideal biases, making a base model happen to "see" the label bias once the initiation operation is done. This phenomenon is empirically held, especially for small-scale classification tasks.

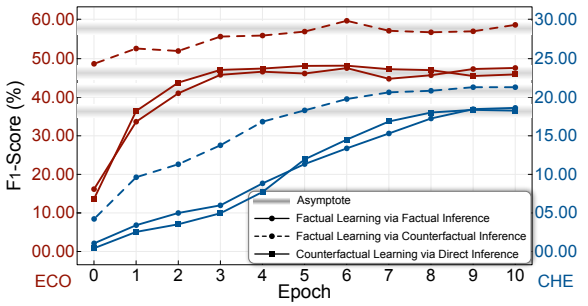

Figure 2: The average results of three types of different learning paradigms on two datasets, including a factual learning with factual inference, a factual learning with counterfactual inference (*i.e.,* CORSAIR) and a counterfactual learning with direct inference.

Surprisingly, *counterfactual learning converges at the factual learning case*. This finding consistently holds on all other baselines across datasets, which means that the so-called counterfactual learning actually degrades to a factual inference. This indicates that if a training model explicitly mitigates two types of dataset biases in an end-to-end fashion, *i.e.,* without the operational gap, it actually loses the function to perform debiased inference. The important reason is that under such circumstance, the potential biases actually "spread" throughout the whole model architecture, instead of the mere part before bias removal is operated, which makes bias removal only look like debiasing but is just a factual feedforward operation that is unable to capture, distill and even mitigate biases. Therefore, the counterfactual inference works only when the operational gap between learning and inferencing exists. This beneficial gap instead makes the biases spread only throughout the part before the bias removal module, and thus enables them to be distilled via counterfactual inference.

4 Related Work

Text classification is a backbone component in many downstream tasks or applications (Broder et al., 2007; Chen et al., 2019; Sun et al., 2019; Qian et al., 2020a,c). Earlier text classification methods focus on manual *feature engineering* (Aggarwal and Zhai, 2012; Cavnar and Trenkle, 1994; Post and Bergsma, 2013). The key factor of *text classification* lies in the quality of *text representation* (Mikolov et al., 2013b,a; Pennington et al., 2014; Canuto et al., 2019; Yan, 2009; Qian et al., 2021). Benefiting from high-quality word vectors, some subsequent studies explored different types of downstream text classification models, including support vector machine (Joachims, 1999), maximum entropy model (Nigamy and McCallum, 1999), naive Bayes (Pang et al., 2002), word clustering (Baker and McCallum, 1998) and neural networks (Kim, 2014; Zhou et al., 2016; Howard and Ruder, 2018; Devlin et al., 2019; Liu et al., 2019).

To solve the dataset bias issue, a straightforward solution is to perform data-level manipulations to prevent models from capturing the unintended dataset biases in model training, including data balance (Dixon et al., 2018; Geng et al., 2007; Chen et al., 2017; Sun et al., 2018; Rayhan et al., 2017; Nguyen et al., 2011) (*a.k.a. resampling*) and data augmentation (Wei and Zou, 2019; Qian et al., 2020b). Another common paradigm for text classification is typically to design model-level balancing mechanisms, including unbiased embedding (Bolukbasi et al., 2016; Kaneko and Bollegala, 2019), threshold correction (Kang et al., 2020; Provost, 2000; Calders and Verwer, 2010) and instance weighting (Zhang et al., 2020; Zhao et al., 2017; Jiang and Zhai, 2007).

5 Conclusion

We have designed a counterfactual framework for text classification debiasing. Extensive experiments demonstrated the framework's good effectiveness, generalizability and fairness. Future work will design a joint-learning technique to dynamically decide each document's main content. We hope the paradigm can illuminate a promising technical direction of causal inference in natural language processing.

Acknowledgements

We thank the anonymous reviewers for their encouraging feedbacks. The work was supported by the National Key Research and Development Program of China (No. 2019YFB1704003), the National Nature Science Foundation of China (No. 71690231), Tsinghua BNRist, Alibaba DAMO academy, NExT++ Research Center and Beijing Key Laboratory of Industrial Bigdata System and Application.

References

Charu C. Aggarwal and ChengXiang Zhai. 2012. A Survey of Text Classification Algorithms. In *Mining Text Data*, pages 163–222.

L. Douglas Baker and Andrew Kachites McCallum. 1998. Distributional Clustering of Words for Text Classification. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 96–103.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (Technology) is Power: A Critical Survey of Bias in NLP. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5454–5476.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *the Conference on Neural Information Processing Systems (NeurIPS)*, pages 4356–4364.

Andrei Broder, Marcus Fontoura, Evgeniy Gabrilovich, et al. 2007. Robust Classification of Rare Queries Using Web Knowledge. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 231–238.

Toon Calders and Sicco Verwer. 2010. Three Naive Bayes Approaches for Discrimination-free Classification. In *Data Mining and Knowledge Discovery*, pages 277–292.

Sergio Canuto, Thiago Salles, et al. 2019. Similarity-Based Synthetic Document Representations for Meta-Feature Generation in Text Classification. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 355–364.

Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Graham Murdock, and Fabrizio Silvestri. 2007. Know Your Neighbors: Web Spam Detection using the Web Topology. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 423–430.

William B Cavnar and John M Trenkle. 1994. N-gram-based Text Categorization. In *Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*.

XiaoShuang Chen, SiSi Li, and MengChu Zhou. 2017. A Noise-Filtered Under-Sampling Scheme for Imbalanced Classification. In *IEEE Transactions on Cybernetics*, pages 4263–4274.

Zhenpeng Chen, Sheng Shen, Ziniu Hu, et al. 2019. Emoji-Powered Representation Learning for Cross-Lingual Sentiment Classification. In *the World Wide Web Conference (WWW)*, pages 251–262.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and Mitigating Unintended Bias in Text Classification. In *the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pages 67–73.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2020. CausaLM: Causal Model Explanation Through Counterfactual Language Models. In *arXiv:2005.13407*.

Guang-Gang Geng, Chun-Heng Wang, Qiu-Dan Li, Lei Xu, and Xiao-Bo Jin. 2007. Boosting the Performance of Web Spam Detection with Ensemble Under-Sampling Classification. In *the Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 583–587.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6904–6913.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8342–8360.

Haibo He and Edwardo A. Garcia. 2009. Learning from Imbalanced Data. In *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, pages 1263–1284.

Chris Hokamp and Qun Liu. 2017. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1535–1546.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 328–339.

Xiaolei Huang and Michael J. Paul. 2018. Examining Temporality in Document Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 694–699.

Xiaolei Huang, Michael C. Smith, Michael J. Paul, Dmytro Ryzhkov, Sandra C. Quinn, David A. Broniatowski, and Mark Dredze. 2017. Examining Patterns of Influenza Vaccination in Social Media. In *the AAAI Conference on Artificial Intelligence (AAAI)*, pages 4–5.

Jing Jiang and ChengXiang Zhai. 2007. Instance Weighting for Domain Adaptation in NLP. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 264–271.

Thorsten Joachims. 1999. Transductive Inference for Text Classification using Support Vector Machines. In *the International Conference on Machine Learning (ICML)*, pages 200–209.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the Evolution of a Scientific Field through Citation Frames. In *Transactions of the Association for Computational Linguistics (TACL)*, pages 391–406.

Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving Debiasing for Pre-trained Word Embeddings. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1641–1650.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, and Jiashi Feng. 2020. Decoupling Representation and Classifier for Long-tailed Recognition. In *the International Conference on Learning Representations (ICLR)*.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 Task 4: Hyperpartisan News Detection. In *the International Workshop on Semantic Evaluation*, pages 829–839.

Kang-Min Kim, Yeachan Kim, Jungho Lee, et al. 2019. From Small-scale to Large-scale Text Classification. In *the World Wide Web Conference (WWW)*, pages 853–862.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *arXiv:1412.6980*.

Jens Kringelum, Sonny Kim Kjaerulff, Soren Brunak, Ole Lund, Tudor I Oprea, and Olivier Taboureau. 2016. ChemProt-3.0: A Global Chemical Biology Diseases Mapping. In *Database (Oxford)*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification. In *the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2267–2273.

Ken Lang. 1995. Newsweeder: Learning to Filter Netnews. In *the International Conference on Machine Learning (ICML)*, pages 331–339.

Frederick Liu and Besim Avci. 2019. Incorporating Priors with Feature Attribution on Text Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6274–6283.

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. In *arXiv:1907.11692*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities and Relations and and Coreference for Scientific Knowledge Graph Construction. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3219–3232.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. In *arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, et al. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *the Conference on Neural Information Processing Systems (NeurIPS)*, pages 3111–3119.

Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling Fairness and Bias in Dynamic Learning-to-Rank. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 429–438.

Hien M. Nguyen, Eric W. Cooper, and Katsuari Kamei. 2011. Borderline Over-Sampling for Imbalanced Data Classification. In *the International Journal of Knowledge Engineering and Soft Data Paradigms (IJKESDP)*, pages 4–21.

Kamal Nigamy and Andrew McCallum. 1999. Using Maximum Entropy for Text Classification. In *the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 61–67.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual VQA: A Cause-Effect Look at Language Bias. In *the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs Up: Sentiment Classification using Machine Llearning Techniques. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86.

Judea Pearl. 2013. Direct and Indirect Effects. In *arXiv:1301.2300*.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. Causal Inference in Statistics: A Primer. In *John Wiley and Sons*.

Judea Pearl and Dana Mackenzie. 2018. The Book of Why: The New Science of Cause and Effect. In *Basic Books*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matt Post and Shane Bergsma. 2013. Explicit and Implicit Syntactic Features for Text Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 866–872.

Foster Provost. 2000. Machine Learning from Imbalanced Data Sets 101. In *the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1–3.

Chen Qian, Fuli Feng, Lijie Wen, Zhenpeng Chen, Li Lin, Yanan Zheng, and Tat-Seng Chua. 2020a. Solving Sequential Text Classification as Board-Game Playing. In *the AAAI Conference on Artificial Intelligence (AAAI)*, pages 8640–8648.

Chen Qian, Fuli Feng, Lijie Wen, and Tat-Seng Chua. 2021. Conceptualized and Contextualized Gaussian Embedding. In *the AAAI Conference on Artificial Intelligence (AAAI)*.

Chen Qian, Fuli Feng, Lijie Wen, Li Lin, and Tat-Seng Chua. 2020b. Enhancing Text Classification via Discovering Additional Semantic Clues from Logograms. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1201–1210.

Chen Qian, Lijie Wen, Akhil Kumar, Leilei Lin, Li Lin, Zan Zong, Shuang Li, and Jianmin Wang. 2020c. An approach for process model extraction by multi-grained text classification. In *Proceedings of The 32nd International Conference on Advanced Information Systems Engineering (CAiSE)*, pages 268–282.

Farshid Rayhan, Sajid Ahmed, Asif Mahbub, Rafsan Jani, Swakkhar Shatabda, and Dewan Md. Farid. 2017. CUSBoost: Cluster-Based Under-Sampling with Boosting for Imbalanced Classification. In *the International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pages 1–5.

Bo Sun, Haiyan Chen, Jiandong Wang, and Hua Xie. 2018. Evolutionary Under-Sampling based Bagging Ensemble Method for Imbalanced Data Classification. In *Frontiers of Computer Science*, pages 331–350.

Lihua Sun, Junpeng Guo, and Yanlin Zhu. 2019. Applying Uncertainty Theory into the Restaurant Recommender System based on Sentiment Analysis of Online Chinese Reviews. In *the World Wide Web Conference (WWW)*, pages 83–100.

Chris Sweeney and Maryam Najafian. 2020. A Transparent Framework for Evaluating Unintended Demographic Bias in Word Embeddings. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1662–1667.

Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. Unbiased Scene Graph Generation from Biased Training. In *the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3716–3725.

Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. 2020. Visual Commonsense R-CNN. In *the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10760–10770.

Zeerak Waseem and Dirk Hovy. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 88–93.

Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6382–6388.

Liuyu Xiang and Guiguang Ding. 2020. Learning From Multiple Experts: Self-paced Knowledge Distillation for Long-tailed Classification. In *the European Conference on Computer Vision (ECCV)*.

Jun Yan. 2009. Text Representation. In *Encyclopedia of Database Systems*.

Xu Yang, Hanwang Zhang, and Jianfei Cai. 2020. Deconfounded Image Captioning: A Causal Retrospect. In *arXiv:2003.03923*.

Guanhua Zhang, Bing Bai, Junqi Zhang, Kun Bai, Conghui Zhu, and Tiejun Zhao. 2020. Demographics Should Not Be the Reason of Toxicity: Mitigating Discrimination in Text Classifications with Instance Weighting. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4134–4145.

Yongfeng Zhang, Guokun Lai, et al. 2014. Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 83–92.

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 *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2979–2989.

Peng Zhou, Wei Shi, Jun Tian, et al. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 207–212.