Replication of “Counterfactual Fairness in Text Classification through Robustness”

Thomas Zeng, Nathan Hedgecock, Jared Chen, Teagan Johnson

Carleton College

Abstract

Deep learning is a common method to create models for the binary task of classifying comments on online forums as toxic or nontoxic. While these models have good performance overall, they often have an inherent bias to classify sentences containing certain identities e.g. “gay”, as toxic [3]. To solve this problem, a method has been proposed to both quantify this phenomenon and counteract this bias [4]. Namely, it proposes the Counterfactual Token Fairness metric and the Counterfactual Logit Pairing loss function. We reimplement the methods of this proposal and evaluate our experimental results against theirs. We also test the robustness of the results with regards to model complexity and the dataset splits used. Our final results approximate their results in terms of the general trends, however there are discrepancies in our values where Counterfactual Logit Pairing underperforms and augmentation overperforms. This provides evidence that their methodology and results are overall robust but may be somewhat limited in the scope of problems it solves.

1 Introduction

Content moderation is important for insuring online forums are inclusive and welcoming. Due to the large number of comments being generated, it would be ideal to create an automated system that can detect the toxicity of online comments. One natural method to do so would be to build a text classifier using deep learning. However, while such text classifiers can yield good performance overall, they often have an inherent bias to classify sentences with certain identities as toxic [3].

As an illustrative example, Table 1 presents predictions in four sentences where only the identity term is changed. The model used was a deep learning model trained in a standard way without utilizing any methods to counteract the inherent bias. We see that the sentences “Some people are **straight**” and “Some people are **Christian**” are predicted by the model as highly likely nontoxic while “Some people are **gay**” is predicted as highly likely toxic. This behaviour is problematic, as it suggests that sentences that contain the word “gay” in them have a higher chance of being flagged as toxic by virtue of containing that word. It is hypothesized that this problem results from imbalanced data where certain identities e.g. “gay”, are primarily present only in toxic comments. This makes the model form a spurious correlation – and thus causing this fairness problem [3].

To quantify and solve this problem, various metrics and training methods have been developed. We focus on the methods proposed in “Counterfactual Fairness in Text Classification through Robustness” [4] – specifically the Counterfactual Token Fairness (CTF) metric and the Counterfactual Logit Pairing

Table 1: Model predictions (as probability of being toxic) on four sentences where only the identity term is perturbed.

Sentence	Model Toxicity Prediction
Some people are **straight**.	0.03
Some people are **gay**.	0.99
Some people are **black**.	0.47
Some people are **Christian**.	0.02

(CLP) loss term. The original paper tests the performance of the CLP loss term against various simple data preprocessing techniques. We replicate their experiments to test the validity and robustness of their methodology.

2 Related Work

As our work is primarily a replication study, it uses the same methodology as seen in [4] – which we will further describe in section 4. The methodology of this paper is built upon the work of [3] – specifically using the same set of identity terms and similar datasets for training and evaluation. However [3] has a greater emphasis on evaluating the fairness of the classifier while [4] focuses more on training models invariant to counterfactual alteration.

The application of counterfactual examples to improve classifier fairness is originally proposed in [7]. Specifically, counterfactual fairness captures the intuition that a model prediction is fair if it is the same across the actual world and a "counterfactual" world where the individual belongs to a different demography. The paper we implement extends this work by narrowing the scope of counterfactual fairness specifically to the task of comment toxicity classification.

The proposed methodology for CLP is inspired by adversarial logit pairing [5]. The methods proposed in [5] are for counteracting adversarial attacks by training classifiers that output similar logits for any given input and its adversarial counterpart e.g. an image and an imperceptibly perturbed version of it that causes it to be misclassified. This methodology is adapted for the fairness problem by replacing the adversarial counterpart with a counterfactual counterpart.

3 Problem Formalization

3.1 Counterfactual Fairness

We now formalize the phenomenon that was illustrated in Table 1 as the problem of *counterfactual fairness*. Specifically, given a classifier f and dataset X, we say f is *counterfactually fair* with regards to a counterfactual generation function Φ and error rate ϵ if

$$|f(x) - f(x')| \leq \epsilon, \quad \forall x \in X, x' \in \Phi(x). \tag{1}$$

The counterfactual generation function Φ is a function that maps a sentence x to a set of "counterfactuals" of that sentence. The exact definition of the "counterfactual" of a sentence can vary depending on the specific fairness we want to test in our model. In the next subsection, we will further narrow the type of counterfactual fairness we are evaluating to counterfactual *token* fairness (CTF), as proposed by the original paper. We do this by defining a specific counterfactual generation function where the "counterfactuals" of a sentence are given by perturbing an identity word in the sentence.

3.2 Counterfactual Token Fairness Metric

As described previously, we have a formal definition for counterfactual fairness, namely equation 1. We now define the counterfactual generation function Φ for CTF.

In the case of CTF, Φ is denoted $\Phi_{\mathcal{A}}$ where $\mathcal{A}$ is the set of all identity terms. Given $a, a' \in \mathcal{A}$, $\Phi_{a,a'}$ is defined as the function that swaps all a tokens with a' and vice versa in any example x. With this we can define $\Phi_{\mathcal{A}}$ as:

$$\Phi_{\mathcal{A}}(x) = \bigcup_{a \neq a' \in \mathcal{A}} \Phi_{a,a'}(x).$$

As an illustrative example we use the sentences from Table 1. Let the set of identities be

$$\mathcal{A} = \{\text{straight, gay, black, Christian}\}.$$

We would thus have

$$\begin{aligned}\Phi_{\mathcal{A}}(\text{“Some people are straight”}) = \{&\text{“Some people are gay”},\\ &\text{“Some people are black”},\\ &\text{“Some people are Christian”}\}\end{aligned}$$

for our counterfactual token generation function.

With a formal definition of CTF, we now describe the CTF-gap as our metric. Namely, we average over the difference in model predictions on the sentences in the dataset and their counterfactual counterparts. Formally, we define it as:

$$\text{CTF-GAP} = \sum_{x \in X} \mathbb{E}_{x' \sim \text{Unif}[\Phi_{\mathbb{A}}(x)]} |f(x) - f(x')|.$$

More concretely, if we take table 1 as an example, with "Some people are straight" as the original sentence, we would have for this sentence that

$$\begin{aligned}\text{CTF-GAP} &= \frac{\sum_{i \in \{\text{gay, black, Christian}\}} |f(\text{“ ... straight”}) - f(\text{“ ... } i\text{”})|}{3}\\ &= \frac{|0.03 - 0.99| + |0.03 - 0.47| + |0.03 - 0.02|}{3}\\ &= 0.47.\end{aligned}$$

To then compute the CTF-gap over the entire dataset we simply average the CTF-gaps for each sentence. The closer the CTF-gap is to zero, the closer the toxicity predictions are for each of a sentence x's counterfactuals. Therefore, the closer the CTF-gap is to zero, the more fair the classification for sentence x.

4 Fairness Methods

Our methods to train counterfactually fair models closely follow that of [4] which we describe in this section. Specifically, the original paper proposes three training methods – two of which are simple preprocesssing methods (blindness and counterfactual augmentation) and the other the aforementioned CLP loss. Due to the lack of specific implementation details in the original paper, we also elaborate on the design decisions we have made.

4.1 Data Preprocessing Methods

4.1.1 Blindness

Blindness substitutes all identity tokens with a special, arbitrary `identity` token. This allows the predictor to know that an identity term is present but not which one. This forces the classifier to be invariant to the identity in a sentence by simply removing it.

Some people are **straight**	→	Some people are `identity`
Some people are **gay**	→	Some people are `identity`
Some people are **black**	→	Some people are `identity`
Some people are **Christian**	→	Some people are `identity`

Figure 1: Example of the blind methodology. Left side is our original sentences. Right side is our sentences after blind preprocessing.

This method works well for known identities, but offers no bias correction for identities that are not blinded. Specifically, this identity invariance only holds for the explicit identities that we masked in the original dataset. For any other identities, the model still outputs problematic behaviours. Later in section 5, we demonstrate this phenomenon.

4.1.2 Counterfactual Augmentation

Counterfactual augmentation augments the training set with generated counterfactual examples. For example, if the original dataset contains sentence "I am white", counterfactual augmentation would add the sentences "I am black", "I am ___", etc. to the training set. These generated examples are meant to guide the model to become invariant to perturbing identity terms. Counterfactual augmentation has the potential to perform better on held-out identities than blindness due to the fact that each sentence with a training identity term has 34 generated examples. These examples will place a greater weight on the sentence's underlying structure, allowing the model to better avoid bias with respect to the identity terms.

However there is a trade-off with this method in that there are some so-called *asymmetric counterfactuals* that this method would have bias against. For example while "That is so gay" should have a toxic connotation, "That is so straight" should not. In this case, there is an asymmetry where the presence of "gay" makes the sentence toxic while "straight" does not (this is as opposed to table 1 where changing "straight" to "gay" does not affect the sentence toxicity). Although asymmetric counterfactuals present the same issues in blindness, it is a more significant issue for counterfactual augmentation because counterfactual augmentation generates many sentences with the same potentially incorrect classification.

With augment, we do not account for counterfactual asymmetries, and resultingly it may overgenerate training examples that are incorrect.

4.2 Counterfactual Logit Pairing

Counterfactual Logit Pairing adds a term to the loss function with the intuition of directly optimizing fairness of the model during training. The CLP term is similar to the CTF metric which we will define next. However we use the logit of the classifier rather than the output in order to guide the model towards better internal representations of the sentences. Specifically if we decompose the classifier f as $\sigma(g(x))$ where σ is the final non-linear activation, we say that the CLP loss term is

$$\sum_{x \in X} \mathbb{E}_{x' \sim \text{Unif}[\Phi(x)]} |g(x) - g(x')|.$$

Thus if $\ell(f(x), y)$ is our original loss function, the CLP loss is

$$\ell(f(x), y) + \lambda \sum_{x \in X} \mathbb{E}_{x' \sim \text{Unif}[\Phi(x)]} |g(x) - g(x')|.$$

For tractability, we compute the expected value of the summation by randomly sampling one counterfactual example. CLP may be better than the previous two methods because in contrast to counterfactual augmentation, the robustness term in the CLP loss explicitly guides the model to satisfy two desirable properties: (1) ensuring a model produces similar outputs on counterfactual pairs and (2) learning models that generalize well to different identities.

4.3 Datasets

Following [4], we train our model using a public dataset released by Google Jigsaw for a Kaggle "Toxic Comment Classification" Challenge [2]. The dataset consists of comments pulled from Wikipedia Talk pages with human-rated toxicity labels.

For evaluation of model performance with the CTF metric, [4] uses two datasets. One is a private data set that they state has a high occurence of comments with identity tokens in them. The other is a synthetic dataset from [3] of sentences like "I am a protestant, straight person, and I hate your guts." As the dataset is synthetically created, it provides an artificial environment to better isolate the counterfactual fairness of the model.

The set of identity terms used for training and evaluation is a set of 50 terms that also originally came from [3]. Out of these, 47 are single tokens and three are bigrams. The terms are randomly partitioned into a training set of 35, and an evaluation set of 12. The three bigrams are also included in the evaluation set since blindness cannot address these during training.

4.4 Replication

Due to the lack of implementation details in the original paper it is hard to replicate their methodology exactly. Furthermore as one of their evaluation datasets is private, our numerical results will also differ as we use a publicly available dataset instead. Therefore, we try to be as faithful to their methodology as possible while making suitable replacement for specifications that they do not provide.

The original paper specifies that they use a convolutional neural network (CNN) as their deep learning architecture. As they do not provide any additional detail, we use the CNN architecture as described in [11] with kernel sizes of 2, 3, 4, and 5. We additionally have 300 filters for each kernel size. We use 300 dimensional Word2Vec embeddings [8] pretrained on Google News for our embedding layer.

We use a batch size of 64, cross entropy loss for our loss function and AdamW as our optimizer for each model tested. The hyperparameter are PyTorch default values.

For evaluation data we replace the private dataset with the publicly available Civil Comments dataset [6], under the recommendation of the authors of the original paper. This dataset is a collection of forum comments that have been labeled with a toxicity rating between 0 and 1. The paper used a toxicity threshold of 0.5, so we've done the same.

Lastly, for counterfactual augmentation, we augment the training set with all possible counterfactual for any given example. The counterfactual examples are assigned the same label as the original example. It is not made explicitly clear whether the original paper augmented the data set with all possible counterfactuals, a subset of counterfactuals, or a singular random counterfactual. Hence, for our main experiments we augment our data set with all possible counterfactuals. In appendix **??** we also show results for augmenting with a singular random counterfactual.

5 Results

Figure 2 shows the CTF gaps of various models on examples containing training terms from the synthetic dataset.[1]

By design, both the original paper's blindness model and our blindness model have a zero CTF gap on examples with training identities. We see similar trends in the re-implementation of our various models, but note that our CLP models performs less fairly than theirs, and our augment model performs more fairly on non-toxic examples.

Figure 3 shows the CTF gaps of various models on non-toxic examples from our evaluation sets. We see similar trends between the original paper and our re-implementation: the blind models do not generalize to non-training identities whereas the augment and CLP models do. Our CLP models decrease CTF gaps more than the original paper (relative to baseline), but do so at a cost to accuracy.

Figure 8 shows the True Negative Rate (TNR) and True Positive Rate (TPR) across examples containing training identities. 'Original' values were evaluated on a private dataset while our 'Re-implementation' values were evaluated on the Civil Comments dataset. Re-implementation results have been multiplied by a constant factor in order to normalize our baseline. We recognize the same trend as the original authors: that better CTF gaps come with a trade-off in TPR. However, we see a much sharper decrease in our TPR rates as we increase λ in our CLP model. While our high λ CLP model obtains a near-zero CTF gap, it does so trivially by predicting most examples as non-toxic. The actual ratio of non-toxic to toxic examples containing training identities within the Civil Comments dataset is 0.69.

[1] Non-toxic examples are evaluated separately as they are less likely to contain asymmetric counterfactuals.

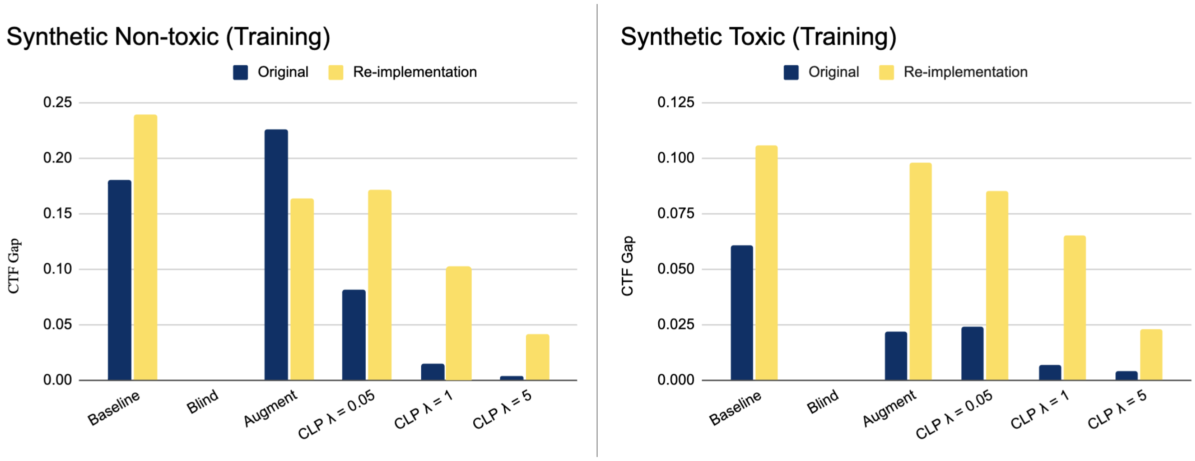

Figure 2: Counterfactual token fairness gaps for non-toxic (Left) and toxic (Right) examples from the synthetic dataset from both the original paper and our re-implementation. All gaps are measured with respect to 35 training terms.

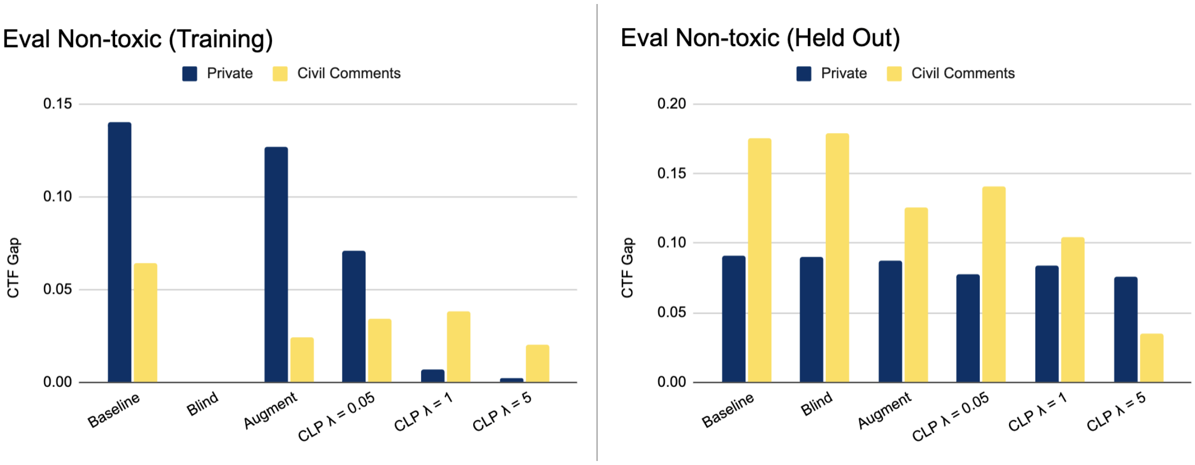

Figure 3: Counterfactual token fairness gaps for non-toxic examples from evaluation sets. ’Private’ values are from [4] and are evaluated on a private dataset while our ’Civil Comments’ values are evaluated on the Civil Comments dataset. Gaps are measured with respect to 35 training terms (Left) and 15 held out terms (Right).

Additional results from experiments testing the robustness of these models can be found in the appendix.

6 Discussion

Both our blind model and the original perform fairly on training identities but do not generalize to held-out terms.

Our augment model performs more fairly than the original (relative to our baseline). One explanation for this could be that the model had more room to improve the fairness of our baseline since our baseline achieved a higher CTF gap. Another possible explanation is that the original paper may have only supplemented a subset of the possible counterfactually perturbed examples for each original input in

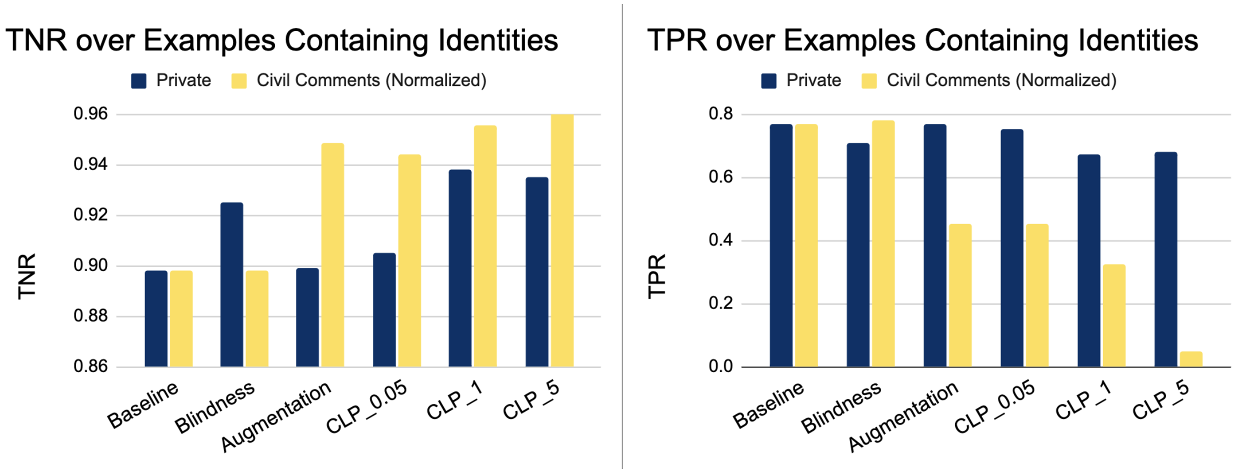

Figure 4: Plot of the average TNR and TPR across examples containing training identities. Re-implementation results have been multiplied by a constant factor in order to normalize our baseline.

their Augmentation model whereas we supplement every possible counterfactually perturbed example. To investigate this hypothesis, we separately evaluate the fairness of a second augment model in which we perturb each example with a single random identity term. In Table 4, we see that our second augment model reduces CTF gaps less than our first model bringing it closer to that of the original paper for examples containing training terms, but further for examples containing held out terms.

Our CLP models simultaneously decrease the CTF gap and increase TNR as the original paper did, but do so at a much higher cost to TPR. We postulate that this is due to the separate evaluation data set that we use. The Civil Comments data set performs notoriously poorly when there is a distribution shift with regard to identities in the train and test sets [6]. We purposefully create this distribution shift by designating a separate train and test split of our identities. In addition, we further compound this issue by training on a separate data set from Civil Comments. A previous analysis on the Civil Comments data set found that certain subgroups have particularly bad accuracy; for example, their baseline model achieved a 57% accuracy on toxic comments containing a religion [6].

From our results, we conclude that the approach proposed by the original paper successfully reduces the CTF gap on different datasets; however, we find a much greater tradeoff to TPR than the original paper. This makes us question whether CLP can be applied generally to various datasets.

6.1 Future Work

Going forward, we would like to improve our methodology for generating counterfactuals. While utilizing only non-toxic comments decreases the number of asymmetric counterfactuals, it does not completely eliminate them. We still find asymmetric examples such as a non-toxic comment containing "black power" being perturbed into a toxic comment containing "white power" [4]. In addition, our counterfactual generation does not take into account occurrences of identity terms not being used as an identity, such as "black car" being replaced with "transgender car."

One possible method to improve our counterfactual generation would be to use word embeddings to group sets of identity pairs such as "(man, woman)" [10]. We could then perturb examples with more relevant identity terms.

We could also consider improving counterfactual fairness via different training methods such as Invariant risk minimization (IRM) which assumes that the training set contains multiple domains, and attempts to find associations which are invariant across those domains. This could be especially useful since our training examples are not uniformly distributed [1]. Several studies have used IRM to successfully improve the fairness of text classifiers across different identities [9, 6].

6.2 Reproducibility

We train and test our models using PyTorch. The code we use to run our experiments can be found at `https://github.com/mtzig/NLP_CTF`.

Acknowledgement

Thank you to Dr. Anna Rafferty of Carleton College for her support and guidance throughout our research. Also thank you to the Carleton College computer science department for allowing us to conduct this research. We would also like to thank the authors of [4].

References

[1] Yo Joong Choe, Jiyeon Ham, and Kyubyong Park. An empirical study of invariant risk minimization. *arXiv preprint arXiv:2004.05007*, 2020.

[2] Lucas Dixon. Toxic comment classification challenge. `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge`, 2017.

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

[4] Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226, 2019.

[5] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[6] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[7] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[9] Qi Qi, Shervin Ardeshir, Yi Xu, and Tianbao Yang. Fairness via adversarial attribute neighbourhood robust learning. *arXiv preprint arXiv:2210.06630*, 2022.

[10] Mohit Wadhwa, Mohan Bhambhani, Ashvini Jindal, Uma Sawant, and Ramanujam Madhavan. Fairness for text classification tasks with identity information data augmentation methods. *arXiv preprint arXiv:2203.03541*, 2022.

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

A Additional Results

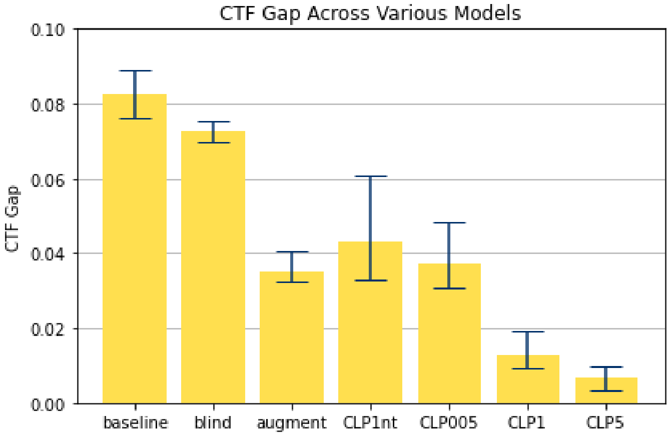

Figure 5: Plot of the average CTF Gap across examples containing identity terms for various models. Error bars were determined by running 3 trials with different random identity splits.

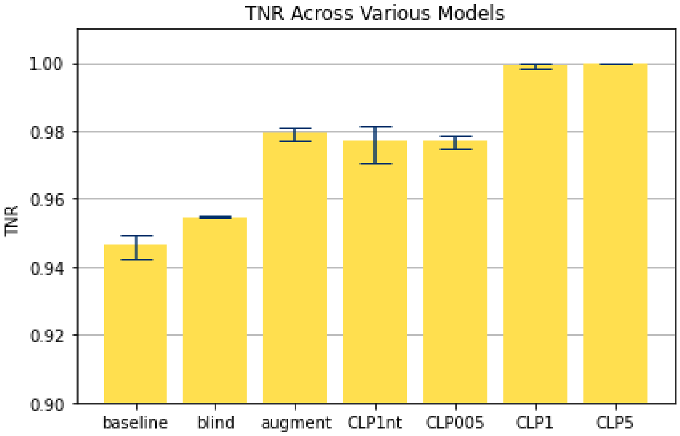

Figure 6: Plot of the average TNR across examples containing identity terms for various models. Error bars were determined by running 3 trials with different random identity splits.

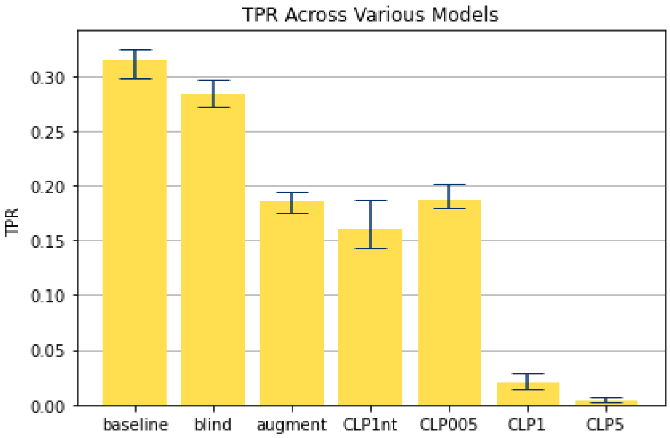

Figure 7: Plot of the average TPR across examples containing identity terms for various models. Error bars were determined by running 3 trials with different random identity splits.

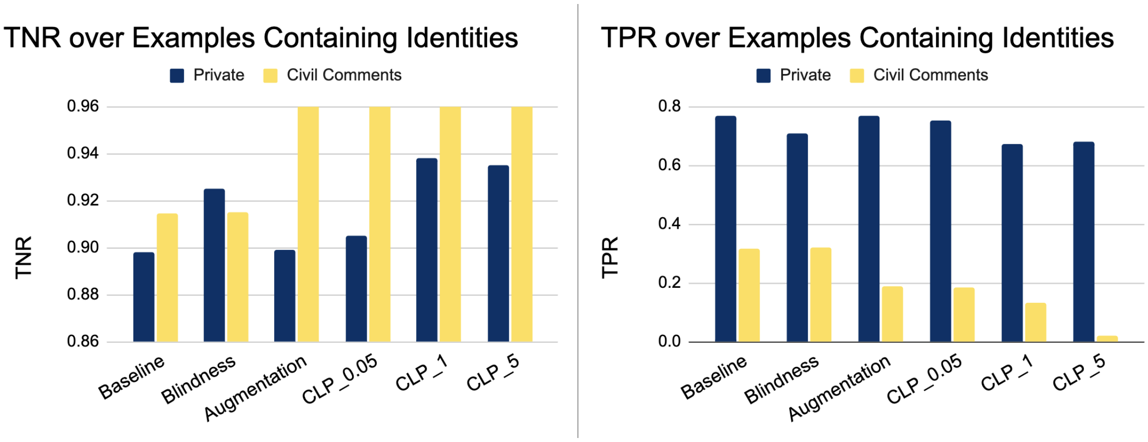

Figure 8: Plot of the average TNR and TPR across examples containing training identities.

Table 2: Counterfactual token fairness gaps for non-toxic (NT) examples from the authentic evaluation data set and both non-toxic and toxic (Tox) examples from a synthetic test set. 'Orig.' values are from [4], and 'Reimp.' values are from our re-implementation. All gaps are measured with respect to 35 training terms. Private is the private dataset used by the original paper for evaluation and C.C. is the Civil Comments dataset we used.

	Eval. NT		**Synth NT**		**Synth Tox**	
	Private	**C.C.**	Orig.	**Reimp.**	Orig.	**Reimp.**
Baseline	0.140	0.064	0.180	0.239	0.061	0.106
Blind	0.000	0.000	0.000	0.000	0.000	0.000
Augment	0.127	0.024	0.226	0.164	0.022	0.098
CLP_nontox $\lambda = 1$	0.012	0.0301	0.015	0.0916	0.007	0.060
CLP $\lambda = 0.05$	0.071	0.034	0.082	0.171	0.024	0.085
CLP $\lambda = 1$	0.007	0.038	0.015	0.102	0.007	0.065
CLP $\lambda = 5$	0.002	0.020	0.004	0.041	0.004	0.023

Table 3: CTF gaps on held out identity terms for non-toxic examples from the evaluation set for both the original paper on the private dataset and our re-implementation on Civil Comments.

	Eval. Held Out	
	Private	**C.C.**
Baseline	0.091	0.175
Blind	0.090	0.179
Augment	0.087	0.126
CLP_nontox $\lambda = 1$	0.095	0.083
CLP $\lambda = 0.05$	0.078	0.141
CLP $\lambda = 1$	0.084	0.104
CLP $\lambda = 5$	0.076	0.035

Table 4: CTF gaps for augment model which perturbs only a single identity term per training example. The model is evaluated separately on non-toxic training and held out terms from the Civil Comments (CC) dataset as well as non-toxic (NT) and toxic (Tox) training terms from the synthetic (Synth) dataset.

CC Train	CC Held Out	Synth NT	Synth Tox
0.056	0.161	0.210	0.102

Table 5: Toxicity probability predictions across models trained using different proposed methods

	Orig. Base	Reimp. Base	Reimp. Blind	Reimp. CF Aug	Reimp. CLP $\lambda = 5$
Some people are gay	0.98	0.987	0.998	0.988	0.062
Some people are straight	0.02	0.025	0.008	0.021	0.060
Some people are Jewish	0.28	0.042	0.050	0.123	0.065
Some people are Muslim	0.46	0.123	0.083	0.054	0.061
Some people are Christian	0.04	0.019	0.018	0.035	0.062