

# Style Pooling: Automatic Text Style Obfuscation for Improved Classification Fairness

Fatemehsadat Mireshghallah, Taylor Berg-Kirkpatrick

University of California San Diego,  
{fatemeh, tberg}@ucsd.edu

## Abstract

Text style can reveal sensitive attributes of the author (e.g. race or age) to the reader, which can, in turn, lead to privacy violations and bias in both human and algorithmic decisions based on text. For example, the style of writing in job applications might reveal protected attributes of the candidate which could lead to bias in hiring decisions, regardless of whether hiring decisions are made algorithmically or by humans. We propose a VAE-based framework that obfuscates stylistic features of human-generated text through style transfer *by automatically re-writing the text itself*. Our framework operationalizes the notion of obfuscated style in a flexible way that enables two distinct notions of obfuscated style: (1) a minimal notion that effectively *intersects* the various styles seen in training, and (2) a maximal notion that seeks to obfuscate by adding stylistic features of *all sensitive attributes* to text, in effect, computing a *union* of styles. Our style-obfuscation framework can be used for multiple purposes, however, we demonstrate its effectiveness in improving the fairness of downstream classifiers. We also conduct a comprehensive study on style pooling’s effect on fluency, semantic consistency, and attribute removal from text, in two and three domain style obfuscation.<sup>1</sup>

## 1 Introduction

Machine learning (ML) algorithms are used in a wide range of tasks, including high-stakes applications like determining credit ratings, setting insurance policy rates, making hiring decisions, and performing facial recognition. It has been shown that such algorithms can produce outcomes that are biased towards a certain gender or race (Buo-lamwini and Gebru, 2018; Silva et al., 2021; Sheng et al., 2021).

Ideally, high-stakes decisions made by either humans or ML algorithms, should not be influenced by irrelevant, protected attributes like nationality,

age, or gender. In many instances, the input data used for making high-stakes decisions is text that is authored by a human candidate – for example, hiring decisions are often based on bios and personal statements. Recent work (De-Arteaga et al., 2019) shows that automatic hiring-decision models trained on bios are less likely to select female candidates for certain roles (e.g. architect, software engineer, and surgeon) even when the gender of the author is not explicitly provided to the system. Bias is, of course, not limited to algorithmic decisions, humans make biased decisions based on text, even when the protected attributes of the author are not explicitly revealed (Pedreshi et al., 2008). Together, these results indicate that both algorithms and humans can (1) decipher protected attributes of authors based on stylistic features of text, and (2) whether consciously or not, be biased by this information.

A large body of prior work has attempted to address *algorithmic bias* by modifying different stages of the natural language processing (NLP) pipeline. For example, Ravfogel et al. (2020) attempt to de-bias word embeddings used by NLP systems, while Elazar and Goldberg (2018) address the bias in learned model representations and encodings. While effective in many cases, such approaches do nothing to mitigate bias in decisions made by humans based on text. We propose a fundamentally different approach. Rather than mitigating bias in learning algorithms that make decisions based on text, we propose a framework that obfuscates stylistic features of human-generated text *by automatically re-writing the text itself*. By obfuscating stylistic features, readers (human or algorithms) will be less able to infer protected attributes that enable bias.

We introduce a novel framework that enables ‘style pooling’: the automatic transduction of user-generated text to a central, obfuscated style. Notions of ‘centrality’ can themselves introduce bias – for example, a system might learn to obfuscate by mapping all text to the dominant style

<sup>1</sup>Code, models, and data is available at <https://github.com/mireshghallah/style-pooling>

Table 1: Example Blog sentences transformed with A4NT (Shetty et al., 2018) and our proposed Intersection and Union obfuscations. Our Intersection obfuscation aims at changing the style such that it does not reflect either teen or adult style. However, the union, tries to reflect both by making changes like adding “...” to the beginning of the sentence (adult style) while keeping the “grr” (teen style). Or by adding exclamation marks at the end of the sentence.

| Age   | Input Sentence (Original Data)          | A4NT (Baseline)                         | Intersection                           | Union                                       |
|-------|---|---|--|---|
| Teen  | <b>grr</b> ... now i get cold quicker . | <b>grr</b> now i get cold lol .         | <b>hmmm</b> ... now i get cold .       | <b>... grr</b> ... now i get cold quicker . |
| Teen  | it was so <b>fricken</b> hilarious .    | it was so <b>boring</b> hilarious .     | it was so <b>utterly</b> hilarious .   | it was so <b>totally</b> hilarious          |
| Adult | well i 've just been <b>too busy</b> .  | well i 've just been <b>kinda fun</b> . | well i 've just been <b>too busy</b> . | well i 've just been <b>too busy</b> .      |
| Adult | these were <b>common phrases</b> .      | these were <b>common teacher</b> .      | these were <b>common</b> .             | these were <b>common ! !</b>                |

seen in its training corpus. This might ‘white-wash’ text, ignoring stylistic features of underrepresented groups in the learned notion of central style. Our framework operationalizes the notion of centrality in a more flexible way: our probabilistic approach allows us to choose between two distinct notions of centrality. First, we define a variant of our model which is incentivized to learn a minimal notion of central style that effectively *intersects* the various styles seen in training. This is achieved through the design of this variant’s probabilistic prior. We further equip this variant with a novel “de-boosting” mechanism, which amplifies the use of words that are less likely to leak sensitive attributes, and de-incentivizes the use of words whose presence might hint at a particular sensitive attribute. Second, we propose an alternative prior that instead incentivizes a maximal notion of style that seeks to obfuscate by adding stylistic features of all protected attributes to text – in effect, computing a *union* of styles. Table 1 shows our intersection and union obfuscation applied to sentences from the Blogs dataset, and highlights the differences between them.

While we propose both these obfuscations in our framework and leave it to the users to choose, it is worth noting that the cognitive process literature shows that when humans are confronted with conflicting biasing information, they tend to form an opinion about the conflicting text, based on their own implicit biases (Richter and Maier, 2017). Therefore, removing sensitive stylistic features may be more effective than combining them. This is also commensurate with our findings, where we observed that intersection more successfully improves the fairness metric (Section 4.2.1).

We extensively evaluate our proposed framework on a wide range of tasks. First, we compare and contrast our “intersection” and “union” obfuscations on a modified version of the Yelp dataset (Shen et al., 2017) where we have created three stylistic domains by deliberately misspelling three disjoint sets of words. We show that our intersection obfuscation successfully removes these

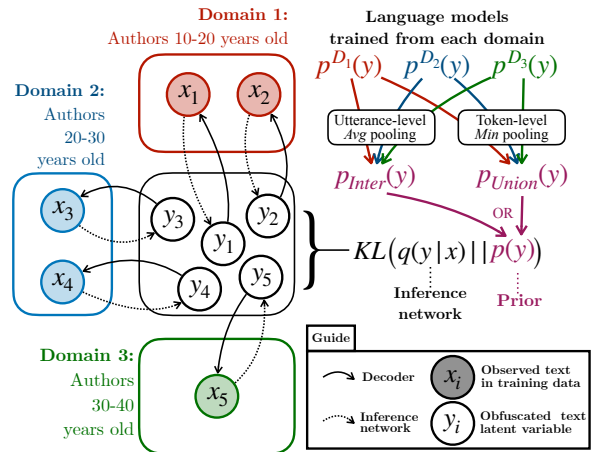


Figure 1: Proposed unsupervised framework for *style pooling*: inducing a centralized obfuscated style.  $x_i$  represent observed text which are clustered by their sensitive attribute (age).  $y_i$  are corresponding latent variables representing the induced obfuscated text. Training leverages an amortized inference setup similar to a VAE-style training, but, critically the prior is produced by pooling language models from each domain using two different strategies targeting (1) intersected style and (2) the union of all styles in the corpus.

misspellings and replaces them by the dominant spelling of the word 99.20% of the time, while our union obfuscation spreads the misspellings into the other two domains 46.40% of the time. Then, we evaluate our framework on the Blogs data (Schler et al., 2006), where the sensitive attribute is age, and we measure the impact our obfuscations have on the fairness of a job classifier, using the the TPR-gap measure from De-Arteaga et al. (2019). We also evaluate the removal of sensitive attributes, fluency of the generated text, and the uncertainty of a sensitive attribute classifier for our framework, in both two and three domain setups.

## 2 Proposed Method

In this section, we first introduce our model structure, then describe our style-pooling priors and the unsupervised learning and inference techniques we leverage for this model class. Finally, we introduce our style de-boosting mechanism.

## 2.1 Model Structure

Consider a training corpus consisting of utterances produced by authors with various protected attributes. In Figure 1, we depict a grouping of authors by age into three domains. We let  $x_i$  represent an individual observed text utterance in the corpus, and assume  $M$  domains (sensitive attribute classes) in the dataset.  $y_i$  is a latent variable that represents the obfuscated version of  $x_i$ . Hence,  $y_i$  is a text valued latent, while  $x_i$  is a text valued observation. We let  $d(i)$  denote the domain of the  $i^{\text{th}}$  sample in the dataset. With this definition, our generative process assumes each sentence  $x_i$ , with corresponding domain  $d(i)$ , is generated as follows: First, a latent sentence  $y_i$  is sampled from a central prior,  $p_{\text{prior}}(y_i)$ , which is domain agnostic. Then,  $x_i$  is sampled conditioned on  $y_i$  from a transduction model,  $p(x_i|y_i; \theta_{y \rightarrow x}^{d(i)})$ . We let  $\theta_{y \rightarrow x}^{D_j}$  represent the parameters of the transduction model for the  $j^{\text{th}}$  domain. We extensively discuss  $p_{\text{prior}}$  in the next section. For now, we assume the prior distributions are pretrained on the observed data and therefore omit their parameters for simplicity of notation. Together, this gives the following joint likelihood:

$$\begin{aligned} p(X^{D_1}, \dots, X^{D_M}, Y; \theta_{y \rightarrow x}^{D_1}, \dots, \theta_{y \rightarrow x}^{D_M}) \\ = \prod_{i=1}^N p(x_i|y_i; \theta_{y \rightarrow x}^{d(i)}) p_{\text{prior}}(y_i) \end{aligned} \quad (1)$$

The log marginal likelihood of the observed data, which we approximate during training, can be written as:

$$\begin{aligned} \log p(X^{D_1}, \dots, X^{D_M}; \theta_{y \rightarrow x}^{D_1}, \dots, \theta_{y \rightarrow x}^{D_M}) \\ = \log \sum_Y p(X^{D_1}, \dots, X^{D_M}; \theta_{y \rightarrow x}^{D_1}, \dots, \theta_{y \rightarrow x}^{D_M}) \end{aligned} \quad (2)$$

**Neural Architectures.** We select a parameterization for our transduction distributions that makes no independence assumptions. We use an encoder-decoder architecture based on the standard attentional Seq2Seq model which has been shown to be successful across various tasks (Bahdanau et al., 2015; Rush et al., 2015). Our prior distributions for each domain are built using recurrent language models which also make no independence assumptions.

## 2.2 Prior Distributions

The critical component of our framework that incentivizes obfuscation are our specialized priors, as depicted in Figure 1. We introduce two prior variants,  $p_{\text{Inter}}(y)$  and  $p_{\text{Union}}(y)$ , which incentivize induction of intersected styles and the union of all styles, respectively. Each prior is assembled out of  $M$  (here  $M = 3$ ) separate language models –  $p^{D_1}, p^{D_2}, \dots, p^{D_M}$  – each trained on the corresponding domain

of observed utterances in the training data. The intersection prior,  $p_{\text{Inter}}(y)$ , is computed by taking the sum of the likelihoods of an entire utterance across the language models from all  $M$  domains (and then re-normalizing to ensure that resulting prior is a valid distribution). This utterance-level average pooling approach incentivizes a ‘‘majority-voting’’ effect, in which the model is pressured to remove any words and stylistic features that are characteristic of one domain, but not the others, and converge to features that are shared by the majority of the domains. Therefore the prior for intersection becomes:

$$p_{\text{Inter}}(y_i) = \frac{1}{M} \sum_j^M p^{D_j}(y_i) \quad (3)$$

In contrast, the union prior,  $p_{\text{Union}}(y)$ , computes the likelihood of an utterance according to the minimum likelihood across each domain’s language model at each token position,  $t$ .<sup>2</sup> Through experimentation (Sec. 4.1) we empirically observed that this prior rewards the model for inserting as many stylistic features as possible that are unique to each domain.

$$p_{\text{Union}}(y_i) \propto \prod_t^T \min(p^{D_1}(y_{i,t}|y_{i,<t}), \dots, p^{D_M}(y_{i,t}|y_{i,<t})) \quad (4)$$

## 2.3 Learning and Inference

Training is accomplished using an approach from (He et al., 2020): We employ seq2seq inference networks and use an amortized inference scheme similar to that used in a conventional VAE, but for sequential discrete latents.

Ideally, learning should directly optimize the log data likelihood, which is the marginal shown in Eq. 2. However, due to our model’s neural parameterization, the marginal is intractable. To overcome the intractability of computing the true data likelihood, we adopt amortized variational inference (Kingma and Welling, 2013) to derive a surrogate objective for learning the evidence lower bound (ELBO) on log marginal likelihood:

$$\begin{aligned} \log p(X^{D_1}, \dots, X^{D_M}; \theta_{y \rightarrow x}^{D_1}, \dots, \theta_{y \rightarrow x}^{D_M}) \\ \geq \mathcal{L}_{\text{ELBO}}(X^{D_1}, \dots, X^{D_M}; \theta_{y \rightarrow x}^{D_1}, \dots, \theta_{y \rightarrow x}^{D_M}, \phi_{x \rightarrow y}) \\ = \sum_i^N \left[ \underbrace{\mathbb{E}_{q(y_i|x_i; \phi_{x \rightarrow y})} [\log p(x_i|y_i; \theta_{y \rightarrow x}^{d(i)})]}_{\text{Reconstruction likelihood}} \right. \\ \left. - \underbrace{D_{\text{KL}}(q(y_i|x_i; \phi_{x \rightarrow y}) || p_{\text{prior}}(y_i))}_{\text{KL regularizer}} \right] \end{aligned} \quad (5)$$

This new objective introduces  $q(y|x; \phi_{x \rightarrow y})$ , which represents the inference network distribution

<sup>2</sup>The token-wise min of the language models is not, itself, a normalized distribution. However, we can treat it as implicitly normalized in our training objective (discussed in the next section) because the absence of normalization only contributes an additive constant to our objective.

that approximates the model’s true posterior,  $p(y|x; \theta_{x \rightarrow y})$ . Learning operates by optimizing the lower bound over both variational and model parameters. Once training is over, the posterior distribution can be used for style obfuscation.

The reconstruction and KL terms in Eq. 5 involve intractable expectations, which means we need to approximate their gradients. To address this, we use the Gumbel-softmax (Jang et al., 2017) straight-through estimator to backpropagate gradients from both the KL and reconstruction loss terms.

**Length Control.** During the training of the model, we observed that it tends to repeat the same word when it is trying to generate obfuscated text,  $y_i$ . To mitigate this, we append two floating point length tokens to the input of the inference networks decoder at each step  $t$ , one of these tokens tells the model which step it is on, and the other tells it how many steps are left (Kikuchi et al., 2016; Hu et al., 2017). We also experimented with positional embeddings instead of floating point tokens, but we observed that they yield worse convergence. Another measure we take to encourage shorter sentences was to hard stop the decoding during training once the re-written sentence had the same length as the original sentence. To further stabilize training we share parameters between the inference network and the transduction models, appending an embedding to the input to indicate the output domain.

## 2.4 Style De-boosting

To better encourage the removal of identifying stylistic features, we introduce a de-boosting mechanism, which incentivizes the use of words that are less likely to leak sensitive attributes, and de-incentivizes the use of words whose presence might hint at a particular sensitive attribute. We build on the intuition that for a given word  $w$  in the vocabulary, if the probability that it belongs to domain  $m$  is similar to the probability that it belongs to domain  $k$ , for any given  $m, k$  within the possible domains,  $M$ , then we can assume that this word does not reveal style. However, if there is a huge gap in the two probabilities, that word might hint at a certain domain if it is present in the re-written text. Therefore, we devise a normalized “style score”,  $s$ , for each word  $w$  in the vocabulary<sup>3</sup>:

$$s_w = \frac{\max(f_w^{D_1}, f_w^{D_2}, \dots, f_w^{D_M}) - \min(f_w^{D_1}, f_w^{D_2}, \dots, f_w^{D_M})}{\max(f_w^{D_1}, f_w^{D_2}, \dots, f_w^{D_M})} \quad (6)$$

Where  $f_w^{D_1}$  is frequency of word  $w$  in the training corpus for domain  $D_1$ , divided by the overall number of tokens (words) in the domain corpus. Using these scores, we modify the output logits of the decoder so that the output probability distribution over the vocabulary for sample  $i$  at step  $t$  is given by:

$$p(y_{i,t} | y_{i,<t}, x_i) \propto \text{softmax}(L_{i,t} - \gamma * S) \quad (7)$$

Here,  $L_{i,t}$  represents the logits at step  $t$ , while  $S$  is the score vector for all the words in the vocabulary.  $\gamma$  is a multiplier that helps tune the amount of de-boosting. Due to the nature of this de-boosting mechanism, it makes sense only to use it with the intersection obfuscation and not the union.

## 3 Experimental Setup

Here, we provide a brief description of our experimental setup. Our code, data and model checkpoints are uploaded in the supplementary material. More details on the code, model configurations, datasets and hyperparameters are provided in Appendix Sections A.1, A.2, A.3 and A.4.

### 3.1 Model Configurations

We used a single layer attentional LSTM-based Seq2Seq encoder-decoder for all the experiments, with hidden layer size of 512 for both encoder and decoder, and word embedding size of 128. For the attribute classifiers and language models, we also use LSTM models with the same architecture, with a final projection layer of the size of sensitive classes/vocabulary.

### 3.2 Datasets

**Synthetic Yelp dataset (Shen et al., 2017).** We shuffle all the sentences in the Yelp reviews dataset and divide them into three groups (domains). We then randomly choose 15 words from the top 20 highest frequency words in the dataset, and allocate the set of top 5 words ( $W_1$ ) to  $D_1$  (domain 1), next 5 to  $D_2$  and the least frequent 5 words to  $D_3$ . We misspell all occurrences of  $W_1$  in  $D_1$ , by changing “word” to “11word11”. We then add “11word11” to the vocabulary, and do this for all the 5 words in all 3 domains (15 words total). After this transformation, we have 3 domains with disjoint stylistic markers, which can help us more concretely analyze our obfuscation mechanism.

<sup>3</sup>While this style score may also highlight *content* that is characteristic of a domain in addition to stylistic word choices, we find in experiments that our use of de-boosting does not substantially harm the utility of downstream classifiers – indicating that content is largely preserved, even with de-boosting.

**Blogs dataset (Schler et al., 2006).** The blogs dataset is a collection of micro blogs containing over 3.3 million sentences along with annotation of author’s age and occupation. We use this data in both two and three domain style pooling, where we treat age as the sensitive attribute and balance the data so each domain has the same number of sentences. In the two domain setup, we divide the data in two groups of teenagers and adults. In the three style setup, we have three groups of teenagers, young adults (20s) and adults (people in their 30s and 40s). We use this dataset for multiple evaluations including fairness. We compare our obfuscation to that of Shetty et al. (2018) in all evaluations with this data.

**Twitter dataset (Rangel et al., 2016).** We use data from the PAN16 dataset, which contains manually annotated (from LinkedIn) age and gender of 436 Twitter users, along with up to 1000 tweets from each user. We use this data for the purpose of sensitive attribute (age) removal comparison with Elazar and Goldberg (2018) in Section A.7, and have therefore used the exact same preprocessing and handling of the data as done by them.

**DIAL dataset (Blodgett et al., 2016).** This is a Twitter dataset which has binary dialect annotations of African American English (AAE) and Standard American English (SAE)<sup>4</sup>, setting “author’s race” as the sensitive attribute. We use this dataset for comparison with the work Xu et al. (2019).

### 3.3 Baselines

**One language model prior (One-LM).** This model is an instance of our framework which uses the output distribution of a single language model as the prior. For the Yelp Synthetic data this single LM is trained on the original data which does not have our modifications and would provide the ideal “intersection”, since the original data itself does not have misspellings from any of our synthetic domains and can be considered as central. In the case of the Blogs data where we don’t have any ideal central data which is void of style, we train an age classifier and then choose the sentences from the training set that the classifier missclassifies. We create a new training set with these samples and train a single LM on them, and use it for the prior. The intuition is that if the classifier could not guess the domain, these samples are probably close to the

<sup>4</sup>Using standard for non-AAE might not be the most suitable naming, but we use it hereon given the lack of a better substitute.

notion of centrality we are looking for.

**A4NT (Shetty et al., 2018).** “A4NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation” is the most closely related past work that also attempts to obfuscate text style through automatic re-writing. However, their adversarial approach uses a discriminator network to hide protected attributes simply by mapping the style of one protected category to that of another.

**PATR (Xu et al., 2019).** Privacy Aware Text Rewriting (PATR) is another work close to ours, which removes sensitive attributes through text re-writing using translation and adversarial learning. Unlike style pooling, PATR, targets privacy and is therefore not concerned with the union vs. intersection of sensitive attributes.

**Original.** We include an “original” baseline in our measurements, which shows the value of a given metric if the original un-obfuscated data was used.

### 3.4 Evaluation Metrics

Below we discuss our evaluation metrics, all of which are measured on the test data.

#### 3.4.1 Fairness

**TPR-gap.** We first define a classifier whose main task is to determine if the occupation of an author is student or not, given text from their blog. We set the age of the author as a sensitive attribute, and want to measure the bias in the classifier, given age. We follow (De-Arteaga et al., 2019) and use the “True Positive Rate gap in age” metric. This measure quantifies the bias in a classifier by finding the gap between the true positive rate for each sensitive attribute group (teen vs. adult). For a binary sensitive attribute  $a$  (age) and a true class (for the classifier’s main task)  $y$ , we define:

$$TPR_{a,y} = P(\hat{Y} = y | A = a, Y = y) \quad (8)$$

$$GAP_{a,y}^{TPR} = TPR_{a,y} - TPR_{a',y} \quad (9)$$

where  $A$  is the random variable denoting binary sensitive attribute with values  $a$  and  $a'$ .  $Y, \hat{Y}$  are random variables denoting the correct class and the predicted class, respectively. The lower the gap is, the more fair the classifier. We report  $GAP_{Teen,Student}$ , which reflects how biased the classifier is towards classifying teens as students.

#### 3.4.2 Linguistic

**Back-Translation (BT) accuracy.** We translate the obfuscated samples back to their original domain using the model, and then for each token

see if it has been correctly back-translated to its origin or not. We use this metric to see whether the obfuscated version contains sufficient information about content to reconstruct the original.

**GPT-2 PPL.** We feed our obfuscated test sentences to a huggingface (Radford et al., 2019) pre-trained GPT-2 medium model, and report its perplexity (PPL), as an automatic measure of fluency. Lower PPL hints at more fluent text.

**BLEU Score.** In the Yelp Synthetic data experiments, since we have the original (not misspelled) text, we can calculate and report the BLEU score.

**GLEU Score.** We use GLEU (Wu et al., 2016) score as another metric for evaluating the fluency of the generated sentences.

**Lexical Diversity (Lex. Div.)** To better quantify the differences between different obfuscations, we calculate the lexical diversity as a ratio where the size of the vocabulary of the model’s output text is the numerator, and the denominator is the overall size of the model’s output text (number of all the tokens in the output).

### 3.4.3 Sensitive-Attribute Classification

**Sensitive-attribute Classifier (Clsf.) accuracy.** To evaluate the removal of sensitive attributes, we train a sensitive-attribute classifier, and use its accuracy as a metric. The closer the accuracy is to chance level (random guess), the more successful is the removal. However, there is a caveat to this metric: it is not always clear how the classifier is making its decision, if it is based on content, or style. Therefore, this metric alone is **not conclusive**.

**Entropy.** To better measure how uncertain the classifier becomes, we also compute its average Entropy across all test samples. Entropy is always between [0.0, 1.0] for two domain classification and [0.0, 1.59] for three domain classification. The higher it is, the more uncertain the classifier is (more desirable for our purpose).

**Confident Response (CR) percentage.** We calculate the percentage of the responses from the classifier for which it was more than 75% sure.

## 4 Experimental Results

### 4.1 Synthetic Yelp Data

Table 2 shows the experimental results for the Synthetic Yelp dataset experiment, where we trained our proposed framework using the three synthetic domains with misspellings, as explained in Section 3.2. The *Corrected*, *Remaining* and *Removed* percentages refer to the average ratio of

Table 2: Results for the Synthetic Yelp dataset with 3 domains. *Corrected* shows what % of modified words in a domain were corrected back to their original format. *Spread* shows the reverse.

|                  | Intersection | Union | One-LM |
|------------------|--------------|-------|--------|
| BT Accuracy (%)  | 92.47        | 94.52 | 95.58  |
| Corrected (%)    | 99.20        | 45.17 | 99.87  |
| Remaining (%)    | 0.61         | 54.37 | 0.00   |
| Removed (%)      | 0.18         | 0.46  | 0.12   |
| Spread (%)       | 0.18         | 46.40 | 0.00   |
| Cls Accuracy (%) | 33.48        | 34.99 | 33.35  |
| BLEU             | 81.74        | 70.86 | 93.01  |

the misspellings corrected, remaining and removed for each domain. These should all sum up to 100%. The *Spread* is the average ratio of the number of words from one domain that have been changed to misspellings from another domain. For instance, if there are 100 occurrences of “word” outside  $D_1$  before obfuscation, if 40 of them are converted to “11word11” after obfuscation, then the spread would be 40%. The *One-LM* can be considered as an “oracle baseline” in this case, since it was trained on original (no misspellings) data.

The main goal of this controlled experiment is to compare and contrast our intersection and union obfuscations. From the Table we can see that both our obfuscations lead to high fidelity (back-translation accuracy) and semantic consistency (BLEU score). They also both render the domain classifier very close to chance level (33.33%). The main differences between these two methods becomes more clear when we look at the corrected, remaining, and spread numbers. The intersection obfuscation with its average pooling, demonstrates a majority voting behavior which incentivizes correcting the misspellings since 2 out of the 3 language models advocate for the correct spelling. Therefore 99.20% of the misspellings are corrected using intersection, very close to the oracle baseline. The Union prior, on the other hand, corrects only 45.17% of the misspellings, and lets 54.37% of them to remain as they are. It also converts 46.40% of the correctly spelled words in other domains to misspellings. This shows that the union is in fact mixing the styles, creating sentences that might have more than one misspelling in them.

### 4.2 Blogs Data

Tables 3, 5 and 6 summarize the experimental results for the Blogs dataset. Below, we will explain each experiment in more detail.

### 4.2.1 Fairness Results

Table 3 shows the results for the fairness metric measurements on text generated using different obfuscations, for “Occupation” classifiers. We have selected a subset of the Blogs data for this experiment, where author occupation is either student or arts, and the age is either teen or adult (two domain obfuscation). We have taken an approach similar to that of Ravfogel et al. (2020), where we create 4 different levels of imbalance. In all cases, the dataset is balanced with respect to both occupation and age. We change only the proportion of each age within each occupation class (e.g., in the 0.8 ratio, the student occupation class is composed of 80% teens and 20% adults, while the arts class is composed of 20% teens and 80% adults). For each imbalance ratio we train the classifier on the original imbalanced data, and then test it with original and automatically generated data from different baselines.

Based on Table 3, we can see that our Intersection obfuscation can improve fairness (TPR-gap) with little harm to the classifier accuracy (Occupation), in comparison to the original data and A4NT. We can trade-off classifier accuracy and fairness, by increasing the de-boosting (DB) multiplier. In the Table, *Intersection* shows Intersection obfuscation with different DB levels. In the case of  $DB = 40$ , we lose slightly more utility, but observe much better fairness.

A4NT’s performance in terms of the fairness metric (TPR-Gap) is comparable to our *Intersection* obfuscation (even without de-boosting), however, in maintaining occupation accuracy (utility), A4NT performs much more poorly. We presume this is because A4NT removes sensitive attributes solely based on hints from a discriminator, and the low occupation accuracy suggests the discriminator captures the content more than it captures style, therefore it changes the meaning and structure of the sentences as well. Our human judgments for semantic consistency and fluency in Section 4.4 support this hypothesis. Our *Union* obfuscation, however, does not improve the fairness. We hypothesize this could be caused by keeping/adding biasing words, which can perpetuate the existing impartialities in the classifier, similar to how human cognition works (Richter and Maier, 2017).

### 4.2.2 Linguistic and Sensitive-attribute Classification Results

The top section of Tables 5 and 6 show the linguistic and sensitive-attribute classification metrics for the

two and three domain obfuscations, respectively. Since A4NT cannot be applied to non-binary style obfuscations as is, there are no results for it in three domains. We can see that for both two and three domains the de-boosting (denoted as DB) offers a trade-off between the linguistic quality of the generated text and the obfuscation of sensitive attributes. Compared to the One-LM baseline, for corresponding levels of de-boosting, our *Intersection* obfuscation is almost always superior, in both text quality and obfuscation. The *Intersection* obfuscation with de-boosting multiplier of 25 outperforms A4NT, with lower classifier accuracy, higher entropy and much lower Confident Response (CR) rate from the classifier. In general, the *Intersection* obfuscation, even without de-boosting does well on *Entropy* and *CR*, which shows that our method is doing well at creating doubt in terms of what the age of the author is. One caveat however, across both two and three domain obfuscations is the classifier accuracy, which does not decrease much. We hypothesize that one reason for this could be the dependency between style and content, and that the sensitive-attribute classifier could be basing its decisions on content, therefore changing the style would not hide the sensitive attribute.

Our *Union* obfuscation is behaving differently from the *Intersection*, and is inferior in terms of obfuscating the text, with higher classifier accuracy and lower entropy. However, it has higher lexical diversity, which could hint at it trying to keep sentences diverse and “adding styles”, whereas the *Intersection* is only keeping the common words and is therefore decreasing the lexical diversity.

## 4.3 Comparison with PATR

Table 4 provides a comparison between our style pooling method, and PATR (Xu et al., 2019).  $\alpha$  is knob used by PATR to tune the intensity of attribute removal, and the classifier accuracy on non-modified text is 86.3%. We can see that without de-boosting, our intersection method drops the classifier accuracy to 74.05% with a GLEU score of 26.32. PATR drops the classifier accuracy to 74.85%, but with a worse level of GLEU. With de-boosting, however, we can achieve a classifier accuracy of 62.12% with GLEU of 17.2, whereas PATR reports accuracy of 65.75% for a much lower GLEU of 9.67 when  $\alpha$  is increased. This shows that our de-boosting mechanism can provide an advantage by giving a lower probability to attribute

Table 3: Fairness results for the Blogs data. The main task is classifying if the author occupation is student or not. Higher occupation accuracy and lower TPR-gap are better. DB denotes our style de-boosting technique, and the number next to it shows its multiplier. Larger multiplier means stronger style obfuscation.

| Ratio | Occupation Accuracy (Utility) |       |              |        |        | TPR-gap (Fairness) |          |      |              |        |        |       |
|-------|-------------------------------|-------|--------------|--------|--------|--------------------|----------|------|--------------|--------|--------|-------|
|       | Original                      | A4NT  | Intersection |        |        | Union              | Original | A4NT | Intersection |        |        | Union |
|       |                               |       | No DB        | DB= 25 | DB= 40 |                    |          |      | No DB        | DB= 25 | DB= 40 |       |
| 0.95  | 74.56                         | 59.35 | 74.77        | 71.12  | 69.73  | 73.22              | 0.54     | 0.29 | 0.23         | 0.23   | 0.21   | 0.51  |
| 0.80  | 65.55                         | 54.74 | 65.31        | 65.12  | 59.60  | 65.43              | 0.35     | 0.21 | 0.35         | 0.18   | 0.18   | 0.36  |
| 0.65  | 59.01                         | 52.73 | 58.41        | 56.68  | 54.45  | 57.19              | 0.12     | 0.05 | 0.11         | 0.11   | 0.11   | 0.15  |
| 0.50  | 58.09                         | 53.47 | 56.21        | 53.6   | 53.18  | 55.49              | 0.04     | 0.08 | 0.05         | 0.05   | 0.03   | 0.05  |

Table 4: Comparison with PATR (Xu et al., 2019), on the Twitter DIAL dataset, where the author’s race is the sensitive attribute.

| Metric        | PATR       |            | Intersection |        | Union |
|---------------|------------|------------|--------------|--------|-------|
|               | $\alpha=1$ | $\alpha=5$ | No DB        | DB= 20 |       |
| GLEU          | 24.77      | 9.67       | 26.32        | 17.21  | 26.25 |
| Clsf. Acc (%) | 74.85      | 65.75      | 74.05        | 62.12  | 73.27 |

revealing components, while maintaining the sentence structure. Our union method also achieves 73.27% accuracy with 26.25 GLEU, making it most suitable for cases where the semantic consistency of the sentences is most important.

#### 4.4 Evaluation with Human Judgments

We design two crowd-sourcing tasks on Amazon Mechanical Turk. (1) Fluency: We provide workers with a pair of obfuscated sentences, and ask them which sentence is more fluent. (2) Semantic Consistency: We provide the original (un-obfuscated) sentences, and ask workers which of the obfuscated sentences is closer in meaning to the original sentence. The model checkpoints used for human evaluations here are those whose fairness and linguistic metrics are reported in Tables 3, 5. We use our intersection obfuscation, with no de-boosting. We randomly select 188 sentences from the test set, and used the model outputs for human judgment. For consistency, each pair of sentences is rated by three workers and we take the majority vote. In terms of fluency, the workers preferred our obfuscations over those of A4NT for 60.38% of the sentences. In terms of semantic consistency, for 72.13% sentences they found our obfuscations to be closer in meaning to the original ones.

## 5 Related Work

A large body of prior work has attempted to address *algorithmic bias* by modifying different stages of the natural language processing (NLP) pipeline. Blodgett et al. (2021), Barikeri et al. (2021), Farrand et al. (2020), Miresghallah et al.

(2021a) and Sheng et al. (2019) propose and analyze benchmarks for evaluating fairness in different applications. Ravfogel et al. (2020), Kaneko and Bollegala (2019), Shin et al. (2020) and Kaneko and Bollegala (2021) attempt to de-bias word embeddings used by NLP systems, while Elazar and Goldberg (2018); Barrett et al. (2019); Wang et al. (2021) attempt to de-bias model representations and encodings.

There is also a large body of work on modifying learning algorithms and inference procedures to produce more fair outcomes (Agarwal et al., 2018; Madras et al., 2018; Zafar et al., 2017; Han et al., 2021; Miresghallah et al., 2021b). While effective in many cases, such approaches do nothing to mitigate *human bias* in decisions based on text. Fundamentally, our framework is concerned with stylistic features of human-generated text. Thus, a large body of prior work on methods for unsupervised style transfer are related to our approach (Santos et al., 2018; Yang et al., 2018; Luo et al., 2019; He et al., 2020). There is also a vast body of work on style obfuscation (Emmery et al., 2018; Reddy and Knight, 2016; Bevendorff et al., 2019; Shetty et al., 2018).

Our work is most closely related to Shetty et al. (2018) and Xu et al. (2019). A4NT (Shetty et al., 2018) attempts to obfuscate text style through automatic re-writing. However, their approach attempts to hide protected attributes simply by mapping the style of one protected category to that of another. In contrast, we seek not to map the author’s text to another author’s style, but to a central obfuscated style. Xu et al. propose Privacy Aware Text Re-writing (PATR), which takes a similar adversarial learning translation based approach to address this problem and re-write text. One fundamental difference between our style-pooling method and PATR is that we provide the choice of union vs. intersection of styles, which is concerned with the societal aspects of removing sensitive attributes, since we

Table 5: Linguistic and sensitive-attribute classifier results for Blogs data, considering *two* sensitive age domains of teens and adults. For BT accuracy and entropy higher is better, for PPL and Confident Response (CR) lower is better.

|        | Metric              | Original | A4NT  | One-LM |         |         | Intersection |         |         | Union |
|--------|---------------------|----------|-------|--------|---------|---------|--------------|---------|---------|-------|
|        |                     |          |       | No DB  | DB = 25 | DB = 40 | No DB        | DB = 25 | DB = 40 |       |
| Ling.  | BT Accuracy (%)     | 100.00   | 66.49 | 94.47  | 92.88   | 90.60   | 95.41        | 87.39   | 88.63   | 96.88 |
|        | GPT-2 PPL           | 41.71    | 44.85 | 39.51  | 53.65   | 66.21   | 41.6         | 42.80   | 58.15   | 42.07 |
|        | Lex. Div. (%)       | 3.22     | 2.28  | 2.52   | 1.82    | 1.09    | 2.50         | 1.47    | 0.97    | 2.71  |
| Clssf. | Clssf. Accuracy (%) | 64.73    | 61.31 | 62.07  | 61.69   | 59.52   | 64.23        | 60.90   | 59.81   | 64.02 |
|        | Entropy             | 0.87     | 0.86  | 0.87   | 0.91    | 0.95    | 0.87         | 0.93    | 0.95    | 0.87  |
|        | CR (%)              | 14.21    | 15.72 | 13.44  | 6.49    | 2.80    | 13.95        | 4.78    | 2.47    | 14.22 |

Table 6: Linguistic and sensitive-attribute classifier results for Blogs data, considering *three* sensitive age domains of teens and adults. For BT accuracy and entropy higher is better, for PPL and Confident Response (CR) lower is better.

|        | Metric              | Original | A4NT | One-LM |         |         | Intersection |         |         | Union |
|--------|---------------------|----------|------|--------|---------|---------|--------------|---------|---------|-------|
|        |                     |          |      | No DB  | DB = 25 | DB = 40 | No DB        | DB = 25 | DB = 40 |       |
| Ling.  | BT Accuracy (%)     | 100.00   | –    | 93.84  | 93.64   | 87.83   | 89.09        | 89.25   | 82.47   | 93.30 |
|        | GPT-2 PPL           | 41.70    | –    | 43.49  | 48.99   | 84.61   | 48.15        | 49.70   | 69.08   | 42.66 |
|        | Lex. Div. (%)       | 3.41     | –    | 2.46   | 1.81    | 0.94    | 1.97         | 1.02    | 0.77    | 2.86  |
| Clssf. | Clssf. Accuracy (%) | 49.78    | –    | 49.16  | 47.64   | 45.41   | 48.12        | 47.13   | 45.81   | 48.81 |
|        | Entropy             | 1.38     | –    | 1.38   | 1.43    | 1.47    | 1.44         | 1.44    | 1.49    | 1.38  |
|        | CR (%)              | 43.00    | –    | 43.33  | 38.89   | 30.75   | 38.76        | 35.37   | 28.02   | 45.88 |

are targeting removal of bias. PATR, however, targets privacy and is therefore not concerned with the union vs. intersection of sensitive attributes.

Finally, there is a body of work on re-writing text to mitigate the potential biases within the content of the text itself. Ma et al. propose PowerTransformer, which rewrites text to correct the implicit and potentially undesirable bias in character portrayals. Pryzant et al. propose a framework that addresses subjective bias in text and Field and Tsvetkov and Zhou et al. introduce approaches to identifying gender bias against women at a comment level and dialect bias in text, respectively. These works focus on the text content, and not on the stylistic features of the author.

## 6 Conclusion

We proposed a probabilistic VAE framework for automatically re-writing text in order to obfuscate stylistic features that might reveal sensitive attributes of the author. We demonstrated in experiments that our proposed framework can indeed reduce bias in downstream text classification. Finally, our model poses two ways of defining a central style. Future work might consider further explorations of alternative notions of stylistic centrality.

## Acknowledgments

The authors would like to thank the anonymous reviewers and meta-reviewers for their helpful

feedback. We also thank Junxian He for insightful discussions. Additionally, we thank our colleagues at the UCSD Berg Lab for their helpful comments and feedback.

## Ethical Considerations

Our proposed model is intended to be used to address a real-world fairness issue. However, this is an extremely complicated topic, and it should be treated with caution, especially upon deploying possible mitigations such as ours. One potential issue we see is the chance that systems like this might obfuscate text by converging towards the majority and erasing styles of marginalized communities. We have tried to address this concern, and raise discussion around it in our introduction and model design, by allowing for multiple operationalizations of a “central” style, and introducing the union and intersection obfuscations. Defining a true notion of centrality that would effectively protect sensitive attributes without erasing any specific styles of writing requires further study.

## References

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR.

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.
- Soumya Barikeri, Anne Lauscher, Ivan Vulić, and Goran Glavaš. 2021. [RedditBias: A real-world resource for bias evaluation and debiasing of conversational 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 (Volume 1: Long Papers)*, pages 1941–1955, Online. Association for Computational Linguistics.
- Maria Barrett, Yova Kementchedjheva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. 2019. Adversarial removal of demographic attributes revisited. In *EMNLP/IJCNLP*.
- Janek Bevendorff, Martin Potthast, Matthias Hagen, and Benno Stein. 2019. Heuristic authorship obfuscation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1098–1108.
- Su Lin Blodgett, Lisa Green, and Brendan O’Connor. 2016. [Demographic dialectal variation in social media: A case study of African-American English](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.
- Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. [Stereotyping Norwegian salmon: An inventory of pitfalls in fairness benchmark datasets](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1004–1015, Online. Association for Computational Linguistics.
- Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of ConNLL*.
- Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR.
- Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128.
- Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21.
- Chris Emmery, Enrique Manjavacas, and Grzegorz Chrupała. 2018. Style obfuscation by invariance. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 984–996.
- Tom Farrand, Fatemehsadat Mireshghallah, Sahib Singh, and Andrew Trask. 2020. Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy. In *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, pages 15–19.
- Anjalie Field and Yulia Tsvetkov. 2020. Unsupervised discovery of implicit gender bias. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 596–608.
- Xudong Han, Timothy Baldwin, and Trevor Cohn. 2021. [Decoupling adversarial training for fair NLP](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 471–477, Online. Association for Computational Linguistics.
- Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. *arXiv preprint arXiv:2002.03912*.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *ICML*.
- Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of ICLR*.
- Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1641–1650.
- Masahiro Kaneko and Danushka Bollegala. 2021. Debiasing pre-trained contextualised embeddings. In *Proceedings of the European Chapter of the Association for Computational Linguistics*.
- Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338.
- Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

- Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. *arXiv preprint arXiv:1905.10060*.
- Xinyao Ma, Maarten Sap, Hannah Rashkin, and Yejin Choi. 2020. Powertransformer: Unsupervised controllable revision for biased language correction. In *EMNLP*.
- David Madras, Toni Pitassi, and Richard Zemel. 2018. Predict responsibly: Improving fairness and accuracy by learning to defer. *Advances in Neural Information Processing Systems*, 31:6147–6157.
- Fatemehsadat Miresghallah, Huseyin Inan, Marcello Hasegawa, Victor Rühle, Taylor Berg-Kirkpatrick, and Robert Sim. 2021a. [Privacy regularization: Joint privacy-utility optimization in LanguageModels](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3799–3807, Online. Association for Computational Linguistics.
- Fatemehsadat Miresghallah, Mohammadkazem Taram, Ali Jalali, Ahmed Taha Taha Elthakeb, Dean Tullsen, and Hadi Esmaeilzadeh. 2021b. Not all features are equal: Discovering essential features for preserving prediction privacy. In *Proceedings of the Web Conference 2021*, pages 669–680.
- Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568.
- Reid Pryzant, Richard Diehl Martinez, Nathan Dass, S. Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *AAAI*.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Francisco Rangel, Paolo Rosso, Ben Verhoeven, Walter Daelemans, Martin Potthast, and Benno Stein. 2016. Overview of the 4th author profiling task at pan 2016: cross-genre evaluations. *Working Notes Papers of the CLEF*, 2016:750–784.
- Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. *arXiv preprint arXiv:2004.07667*.
- Sravana Reddy and Kevin Knight. 2016. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 17–26.
- Tobias Richter and Johanna Maier. 2017. Comprehension of multiple documents with conflicting information: A two-step model of validation. *Educational psychologist*, 52(3):148–166.
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of EMNLP*.
- Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. *arXiv preprint arXiv:1805.07685*.
- Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *Proceedings of AAAI*.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6833–6844.
- Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412.
- Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2021. Societal biases in language generation: Progress and challenges. In *Proceedings of the Conference of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. A4nt: Author attribute anonymity by adversarial training of neural machine translation. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1633–1650, Baltimore, MD. USENIX Association.
- Seungjae Shin, Kyungwoo Song, JoonHo Jang, Hyemi Kim, Weonyoung Joo, and Il-Chul Moon. 2020. Neutralizing gender bias in word embedding with latent disentanglement and counterfactual generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3126–3140.
- Andrew Silva, Pradyumna Tambwekar, and Matthew Gombolay. 2021. [Towards a comprehensive understanding and accurate evaluation of societal biases in pre-trained transformers](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2383–2389, Online. Association for Computational Linguistics.
- Liwen Wang, Yuanmeng Yan, Keqing He, Yanan Wu, and Weiran Xu. 2021. [Dynamically disentangling social bias from task-oriented representations with adversarial attack](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3740–3750, Online. Association for Computational Linguistics.

- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Qionghai Xu, Lizhen Qu, Chenchen Xu, and Ran Cui. 2019. Privacy-aware text rewriting. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 247–257.
- Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. *arXiv preprint arXiv:1805.11749*.
- Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR.
- Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2021. Challenges in automated debiasing for toxic language detection. *arXiv preprint arXiv:2102.00086*.

## A Appendix

### A.1 Experiment Code

We have uploaded code, data and model checkpoints needed for reproducing the experiments in <https://github.com/mireshghallah/style-pooling>. There, you will find a Read Me file which includes all the necessary steps and link to the data and model checkpoints. In short, you need to download the data and model checkpoints from the link. When you download the models-data compressed folder, extract it. Place the content of the data folder and the models in corresponding folders in the code. The package dependencies are all included in the `dependencies` file. In order to create a similar setup, please install the exact version mentioned there.

Once all the directories are setup, you can train your own models using the commands in the Read Me, or you can evaluate the models we have already provided. Evaluation code is included in the `results_ipynbs` folder.

### A.2 Model Configurations

**Seq2Seq Model.** For all the experiments, We use single layer LSTMs with hidden size of 512 as both the encoder and decoder, and we use a word embedding size of 128. We apply dropout to the readout states before softmax with a rate of 0.3. We add a max pooling operation over the encoder hidden states before feeding it to the decoder.

**Language Model: Yelp data.** We use an LSTM language model with hidden size of 512 and word embedding size of 128 and dropout value of 0.3.

**Language Model: Blog and Twitter data.** We use an LSTM language model with hidden size of 2048 and word embedding size of 1024 and dropout value of 0.3.

**Sensitive-attribute Classifiers.** We use LSTM classifiers for classifying sensitive attributes. The hidden size is 512 and word embedding size is 128. The last layer size is the number of sensitive classes.

**GPT-2** We used this repository <https://github.com/priya-dwivedi/Deep-Learning/tree/master/GPT2-HarryPotter-Training> to download and feed data to the GPT-2 model, and get the PPL score.

### A.3 Dataset Details

**Yelp.** The training set contains 399,999 sentences, and test set consists of 30,000 sentences, both divided equally between the 3 domains. The vocabulary size is 9.6k words. The misspelled words of  $D_0$ ,  $D_1$  and  $D_2$  are “00great00, 00this00, 00it00, 00to00, 00food00”, “11of11, 11place11, 11for11, 11good11, 11service11” and “22they22, 22are22, 22in22, 22very22, 22my22”, respectively.

**Blogs.** The blogs dataset is a collection of micro blogs from [blogger.com](http://blogger.com) which consists of 19,320 ‘documents’ (over 3.3 million sentences) along with annotation of author’s age, gender, and occupation. Each document is a collection of an individual author’s posts. We will use this data in both two and three domain style-pooling, where we treat age as the sensitive attribute and balance the data so each domain has the same number of sentences. In the two domain setup, we divide the data in two groups of teenagers, 13 – 18 and adults 23 – 48. In the three style setup, we have three groups of teenagers (13 – 18), young adults (23 – 28) and adults (33 – 48). The age groups 19 – 22 and 29 – 32 are missing from the data. After preprocessing and balancing the dataset, we end-up with 1.2 Million sentences in the training set, 400k sentences in the test for the 2 domain setup, and 762k training sentences and 192k test sentences for the test set. There are 10k words in the vocabulary. All the datasets are balanced.

**Twitter.** There are 146.5k sentences in the training set, and 11.2k sentences in the test set. We reproduced this data using this scripts from [Elazar and Goldberg \(2018\)](https://github.com/yanaiela/demog-text-removal)’s GitHub repository: <https://github.com/yanaiela/demog-text-removal>.

### A.4 Hyperparameters

For all experiments, we set the training batch size to 32, the test batch size to 128 and the temperature of the softmax to 0.01.

**KL weight hyperparameter:** The KL term in Eq. 5 that appears naturally in the ELBO objective, can be treated as a regularizer that uses our  $p_{prior}$  to induce the type of style we want. Therefore, in practice, we add a weight  $\lambda$  to the KL term in ELBO since the regularization strength from our priors varies depending on the datasets, training data size, or prior structures ([Bowman et al., 2016](#)).

**Yelp.** For the Yelp experiments, the learning rate is set to 0.001 and the KL weight ( $\lambda$ ) for the Union, One-LM and Intersection experiments are 0.03, 0.03 and 0.02, respectively.

**Blogs and Twitter.** For the Blogs experiments, the learning rate is 0.0005, and the KL weight ( $\lambda$ ) is 0.04 (for both 2 and 3 domains).

### A.5 Comparison with A4NT Details

To compare with the work “A4NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation” (Shetty et al., 2018), we downloaded a checkpoint of their pre-trained model, available in their github repository: <https://github.com/rakshithShetty/A4NT-author-masking>. Since we have also used the same dataset with the same train/test separation, we use the model as is for evaluation.

### A.6 Human Evaluation Experiment Details

Our crowd workers are recruited from the Amazon Mechanical Turk (AMT) platform. Each HIT required the workers to answer a question regarding only one pair of sentences, and each worker was paid \$0.1 per HIT. For English proficiency, the workers were restricted to be from USA or UK. For the semantic consistency test, the question asked from the Turkers was: “Which sentence is closer in meaning to the original sentence below?”, where the original sentence and the obfuscated ones were provided to the workers. For fluency, we asked: “Which sentence is more fluent in English?”.

### A.7 Comparison with Elazar and Goldberg (2018)

Elazar and Goldberg (2018) aims at creating representations for text that could be used for a specific classification task, while hiding sensitive attributes. Although our approach deals with the text as opposed to representations and can be applied for a wider range of downstream tasks, we offer a brief comparison to this method. Elazar and Goldberg use the Twitter dataset Rangel et al. (2016), set the sensitive attribute to be age, and try to produce representations that would perform well on the main task of “conversation detection” (mention detection) on Tweets. On the original data, they report an accuracy of 77.5% and 64.8% for a classifier that tries to classify conversations and age, respectively, which drop to 72.5% and 57.3%, after applying their adversarial learning scheme.

We cloned their repository and used their code to process the dataset. We then created and trained the conversation and age classifiers, and reached an accuracy of 75.8% and 64.63% for them, respectively. These dropped to 73.28% and 54.2%, after applying applying our intersection method. This shows that for this particular task, our re-written text can out-perform prior work.