Text Style Transfer for Bias Mitigation using Masked Language Modeling

Ewoenam Kwaku Tokpo
Department of Computer Science
University of Antwerp

Toon Calders
Department of Computer Science
University of Antwerp

Abstract

It is well known that textual data on the internet and other digital platforms contain significant levels of bias and stereotypes. Various research findings have concluded that biased texts have significant effects on target demographic groups. For instance, masculine-worded job advertisements tend to be less appealing to female applicants. In this paper, we present a text-style transfer model that can be trained on non-parallel data and be used to automatically mitigate bias in textual data. Our style transfer model improves on the limitations of many existing text style transfer techniques such as the loss of content information. Our model solves such issues by combining latent content encoding with explicit keyword replacement. We will show that this technique produces better content preservation whilst maintaining good style transfer accuracy.

1 Introduction

Authors such as Bolukbasi et al. (2016) and May et al. (2019) have drawn attention to some fairness problems in the NLP domain. In a post on Buzz-Feed (Subbaraman, 2017) with the title, “Scientists Taught A Robot Language. It Immediately Turned Racist", the author reports how various automated language systems are disturbingly learning discriminatory patterns from data. Another prominent example of bias in NLP is Amazon’s AI recruitment tool which turned out to be biased against female applicants (Dastin, 2018). Mitigating bias in textual data before training can be an important preprocessing step in training fair language systems like chatbots, language translation systems, and search engines, but a more direct need for mitigating bias in textual data has been pointed out by various researchers (Gaucher et al., 2011; Tang et al., 2017; Hodel et al., 2017) who have uncovered the worrying issue of bias in job advertisements. This can have significant implications on the job recruitment process. As a matter of fact, Gaucher et al. (Gaucher et al., 2011) explored the effect of biased job advertisements on participants of a survey. They found that changing the wording of a job advertisement to favor a particular gender group considerably reduced the appeal of the job to applicants not belonging to that gender, regardless of the gender stereotype traditionally associated with the job. Consequent to such findings, a few tools and models have been developed to detect and mitigate biases in job advertisements. Some of these tools include text editors like Textio which has been successfully used by companies such as Atlassian to increase diversity in their workforce (Daugherty et al., 2019).

Another area of impact, regarding biased texts, is in news publications; Kiesel et al. (2019) explore the issue of hyperpartisan news from an extreme left or right-wing perspective. Again, with the prevalence of hate speech and microaggression perpetuated on various social media platforms, there have been growing concerns about fairness in such areas.

A machine learning technique that can be employed to mitigate bias in text documents is *style transfer*. Style transfer is a technique that involves converting text or image instances from one domain to another, such that the content and meaning of the instance largely remain the same but the style changes. However, a problem that has challenged research in text style transfer is the relative unavailability of parallel data that would ideally be required to train such models (Rao and Tetreault, 2018; Fu et al., 2018; Shen et al., 2017). Training with parallel data makes it possible to directly map training instances from one domain to the other, hence, facilitating the learning process. Due to this, most style transfer systems mainly employ training techniques that fall under two categories: keyword replacement and auto-encoder sequence-to-sequence techniques. In the case of keyword replacement, biased words are deleted

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, pages 163 - 171
July 10-15, 2022 ©2022 Association for Computational Linguistics

and replaced with alternative words. In the case of the auto-encoder sequence-to-sequence generative approach, the input text is directly encoded by an encoder to get a latent representation of the text, which is subsequently decoded by a decoder.

The main contributions of this work include:

1. The development of an end-to-end text bias mitigation model that can convert a piece of biased text to a neutral version [1] whilst maintaining significant content information. For example, given the female-biased text, *"The event was kid-friendly for all the* ***mothers*** *working in the company"*, our task is to transform this text into a gender-neutral version like *"The event was kid-friendly for all the* ***parents*** *working in the company"*. Our model is trained exclusively on nonparallel data. Since parallel corpora are relatively hard to obtain, training with only non-parallel data is of great importance.

2. A novel way of improving content preservation and fluency in text style transfer by combining keyword replacement and latent content information. Some other key novelties in our work include our approach to generating latent content representation and our approach to identifying attribute tokens.

We make the code and data used in this work available [2].

2 Style transfer

Style transfer has been widely explored in computer vision to convert images from one style to another (Gatys et al., 2016; Huang and Belongie, 2017; Johnson et al., 2016). However, directly applying image style transfer techniques for text is problematic because of the unique characteristics of both domains. For instance, in text, style and content are more tightly coupled and harder to separate (Hu et al., 2020). In addition to that, the non-differentiability of discrete words causes optimization problems (Yang et al., 2018; Lample et al., 2018).

In NLP, style transfer has mostly been explored in areas such as sentiment analysis (Li et al., 2018; Fu et al., 2018; Zhang et al., 2018) and machine translation (Lample et al., 2017). A few style transfer learning techniques use parallel data for training. Hu et al. (2020) give an elaborate survey on such models. In this paper, we will only focus on models that are trained on non-parallel data, some of which we will review in the following subsection.

2.1 Auto-encoder sequence-to-sequence models

Auto-encoder sequence-to-sequence models basically consist of an encoder that encodes the given text into a latent representation which is then decoded by a decoder. Many of these models adopt an adversarial approach to learn to remove any style attribute from the latent representation. The resulting disentangled latent representation is decoded by the decoder in a sequential generative manner.

Shen et al. (2017) propose two models for text style transfer based on the auto-encoder sequence-to-sequence technique: an aligned auto-encoder model and a variant of that, called the cross-aligned auto-encoder model. Prabhumoye et al. (2018) propose a style transfer model using back-translation. This is based on prior research that suggests that language translation retains the meaning of a text but not the stylistic features (Rabinovich et al., 2017).

An issue with Auto-encoder sequence-to-sequence models, in general, is the loss of information due to compression when encoding. Furthermore, Wu et al. (2019) note that sequence-to-sequence models for style transfer often have limited abilities to produce high-quality hidden representations and are unable to generate long meaningful sentences. Nonetheless, sequence-to-sequence generative models can prove more effective in applications where the text needs to be considerably rephrased (eg. from informal style to a formal style).

2.2 Explicit Style Keyword Replacement

These methods follow the general approach of identifying attribute markers, deleting these markers, and predicting appropriate replacements for these markers which conform to the target style. Li et al. (2018) propose the DeleteOnly and the Delete&Retrieve, which use a three-step Delete, Retrieve, and Generate approach. Sudhakar et al. (2019) introduce Blind Generative Style Transformer (B-GST) and Guided Generative Style Transformer (G-GST) as improvements on DeleteOnly and the Delete&Retrieve from (Li et al., 2018).

[1] See Section 7 for discussion on how we define bias.

[2] https://github.com/EwoeT/MLM-style-transfer

Since Explicit Style Keyword Replacement methods only delete a small portion of the input text, they preserve much more information. These systems on the other hand are unable to properly capture information of the deleted tokens (Sudhakar et al., 2019), leading to examples such as *"The event was kid-friendly for all the **mothers** working in the company"* → *"The event was kid-friendly for all the **children** working in the company"*.

3 Methodology

The goal of our model is to transform any piece of biased text into a neutral version. If we take the two style attributes s_a and s_b to represent neutral style and biased style respectively, given a text sample x_b that belongs to s_b, our goal is to convert x_b to x_a, such that x_a belongs to style s_a but has the same semantic content as x_b except for style information.

Our model is composed of four main components, as illustrated in Fig 1. We also illustrate the process with an example in Fig. 2.

3.1 Attribute Masker

The Attribute Masker identifies the attribute words (words responsible for bias in a text) and masks these words with a special *[MASK]* symbol. The resultant text is fed as input to the Token Embedder.

We use LIME (Ribeiro et al., 2016), a model agnostic explainer that can be used on textual data, to identify attribute tokens. Although very effective, using LIME can increase computational time, especially for long text sequences. Some Explicit Style Keyword Replacement models use relatively simple techniques to identify attribute words. Li et al. (2018) use the relative frequency of words in the source style. Others like Sudhakar et al. (2019) employ more advanced methods like using attention weights. However, using techniques like attention weights to identify attribute tokens has been proven to not be very effective (Jain and Wallace, 2019).

To use LIME to detect attribute words, we first need to train a text classifier f that predicts whether a given text is biased. We fine-tune BERT (Devlin et al., 2019), a pretrained language model, as a text classifier by training it on a labeled corpus containing both biased and neutral texts. Lime linearly approximates the local decision boundary of f and assigns weights to tokens based on their influence on the classification outcome. With these weights (scores), we set a threshold value μ to select words to be masked. These words are replaced by a special *[MASK]* token.

3.2 Token Embedder

The Token Embedder is responsible for generating token embeddings for the masked tokens. To do this, we train a BERT model for masked language modeling on a corpus of unbiased texts. The Token Embedder outputs a set of all token embeddings $W = \{w_1, ..., w_n\} \in R^{n \times d}$. Following the convention used by Devlin et al. (2019), we take the size of every embedding to be $d = 768$ throughout this paper.

3.3 Latent-content Encoder

The Latent-content Encoder takes the original (unmasked) text as input and encodes it into a latent content representation. An important part of this stage is our approach to disentangle the resulting latent content representation from the biased style.

The Latent-content Encoder is responsible for generating a latent content representation of the input sentence. For this, we train a BERT embedding model that takes as input the original text (unmasked) x_b and generates a target latent representation $\hat{z}$.

When x_b is given as an input, the Latent-content Encoder first generates token embeddings $v_i \in R^d$ for each token $t_i \in x_b$. The set of token embeddings $V = \{v_1, ..., v_n\} \in R^{n \times d}$ is mean-pooled to generate $\hat{z} \in R^d$. Since we want $\hat{z}$ to have the same content as x_b but not the bias that exists in x_b, we use a dual objective training to debias $\hat{z}$.

Both the Latent-content Encoder and the Source Content Encoder take x_b as input. The Latent-content Encoder generates output $\hat{z}$ whereas the Source Content Encoder generates z. Firstly, the goal is to make $\hat{z}$ and z have the same content, hence, we want them to be as similar as possible. We use the cosine-similarity to quantify this similarity. The similarity loss is minimized using mean-squared error; defined as:

$$\mathcal{L}_{sim} = \frac{1}{N} \sum_{j=1}^{N} (cosine\_similarity(\hat{z}_j, z_j) - 1)^2$$

Secondly, a bias detector takes $\hat{z}$ as input and returns the class probabilities of $\hat{z}$. Because we want $\hat{z}$ to belong to the neutral class, the Latent-content Encoder has to learn to generate $\hat{z}$ that is always classified as neutral. This is achieved by minimizing the cross-entropy loss:

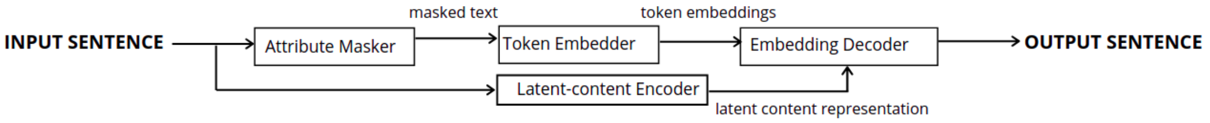

Figure 1: The architecture of our proposed model. The model consists of four main components. The arrows show the flow of information within the model, and how the various components interact with each other.

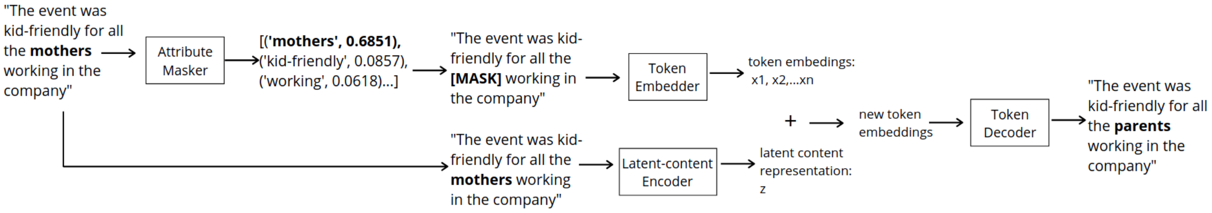
"The event was kid-friendly for all the **mothers** working in the company"
Attribute Masker
[('**mothers**', **0.6851**), ('kid-friendly', 0.0857), ('working', 0.0618)...]
"The event was kid-friendly for all the **[MASK]** working in the company"
Token Embedder
token embedings: x1, x2,...xn
+
new token embeddings
Token Decoder
"The event was kid-friendly for all the **parents** working in the company"
"The event was kid-friendly for all the **mothers** working in the company"
Latent-content Encoder
latent content representation: z

Figure 2: An example to illustrate the end-to-end bias mitigation process. This demonstrates the operation of each component of the model. In the case of multiple attribute words, these attribute words are all masked and replaced simultaneously. The Latent-content Encoder aims to remove traces of gender information from sentence-level semantic content before being added to the token embeddings.

$$\mathcal{L}_{acc_{\hat{z}_j}} = -\sum_{j=1}^{N} logP(s_a|\hat{z}_j)$$

$P(s_a|\hat{z}_j)$ is the classifier's prediction of the probability of $\hat{z}$ being neutral.

Combining both losses we get the dual objective: $LCE\_loss = (1-\lambda)\mathcal{L}_{sim} + \lambda\mathcal{L}_{acc_{\hat{z}_j}}$

3.4 Token Decoder

The Token Decoder computes the average of each token embedding and the latent content representation to generate new token embeddings. The Token Decoder uses these embeddings to predict the correct tokens.

The Token Decoder first adds latent content information to word embeddings. To do this, the Token Decoder takes as inputs both W from the Token Embedder and $\hat{z}$ from the Latent-content Encoder. For each $w_i \in W$, a new token embedding $\hat{w}_i \in R^d$ is generated by computing the weighted average of $w_i \in R^d$ and $\hat{z} \in R^d$. After generating $\hat{w}_i$, the Token Decoder uses it to predict the right token by computing the probability distribution over all the tokens in the vocabulary. We compute the decoding loss as: $\mathcal{L}_{dec} = -\sum_{i=1;t_{\pi_i}\in T_\Pi}^{n} logP(t_{\pi_i}|\hat{w}_{i_\Pi})$

To augment this process, we use a pretrained classifier to ensure that the output sentence x_a is always neutral. A dual objective is again used in this process: $TD\_loss = (1-\gamma)\mathcal{L}_{dec} + \gamma\mathcal{L}_{acc_{x_a}}$. Where $\mathcal{L}_{acc_{x_a}}$ is the loss from the classifier. Because x_a is made up of discrete tokens (one-hot encodings) which are non-differentiable during backpropagation, we use a soft sampling approach as was done in (Wu et al., 2019; Prabhumoye et al., 2018): $t_{\pi_i} \sim softmax(\mathbf{o}_t/\tau)$

4 Experiments

For our experiments, we focus on gender bias (we limit our work to a binary definition of gender) [3]. The use of gender is motivated by the relative availability of resources such as datasets. Nonetheless, we believe that our work is adaptable to other forms of biases such as racial bias since the technique is not dependent on the domain (only neutral and bias examples are needed). To show our technique's applicability in different domains, we experiment on gender obfuscation, where instead of mitigating the bias, we try to convert female-authored texts to "look like" male-authored texts. We arbitrarily chose to convert from female to male just for the sake of experiment; the same technique can be applied for male to female as well.

All experiments are conducted using English language corpus. In the future, we hope to extend our work to cover other languages as well. We discuss the details of our experiments in the following subsections.

4.1 Dataset

We run our experiments [4] on two datasets discussed below. Some statistics of the datasets are given in A Table 3

[3] See Section 7

[4] All experiments are run on a Tesla V100-SXM3 GPU with 32Gb memory.

4.1.1 Jigsaw dataset:

The Jigsaw datasets[5] consists of comments that are labeled by humans with regard to bias towards or against particular demographics. Using the value 0.5 as a threshold, we extract all texts with gender (male or female) label ≥ 0.5 as the gender-biased class of texts and extract a complementary set with gender labels < 0.5 as the neutral class.

4.1.2 Yelp dataset:

We extract this dataset from the preprocessed Yelp dataset used by (Prabhumoye et al., 2018; Reddy and Knight, 2016a). This dataset contains short single sentences which we use for author gender obfuscation.

4.2 Evaluation models and metrics

To evaluate the performance of our model, we compare it to six other models; Delete-only, Delete-and-retrieve (Li et al., 2018), B-GST, G-GST (Sudhakar et al., 2019), CAE (Shen et al., 2017) and BST (Prabhumoye et al., 2018).

The evaluation is based on three automated evaluation metrics for style transfer discussed by Hu et al. (2020); *style transfer accuracy* (Transfer strength), *content preservation*, and *fluency*.

Style transfer accuracy: This gives the percentage of texts that were successfully flipped from the source style (bias style) to the target style (neutral style) by our model. To predict whether a text was successfully flipped, we use a trained BERT classifier different from the one used to train the respective models.

Content preservation: We measure content preservation by computing the similarity between the generated text and the original text. Similar to Fu et al. (2018), we use the cosine similarity between the original text embedding and the transferred text embedding to measure the content preservation. To make this more effective, we generate text embeddings with SBERT (Reimers and Gurevych, 2019), a modified version of pre-trained BERT that generates semantically meaningful sentence embeddings for sentences so that similar sentences have similar sentence embeddings, that can be compared using cosine-similarity.

Fluency: Similar to (Subramanian et al., 2018), we measure the fluency of the generated text using the *perplexity* produced by a Kneser–Ney smoothing 5-gram language model, KenLM (Heafield, 2011) trained on the respective datasets.

Table 1: **Jigsaw dataset-** Transfer strength and Content preservation scores for the models on all three datasets. **C.P.**: Content preservation, **PPL**: Fluency (Perplexity), **Accuracy**: Style transfer accuracy, **Original*:** refers to the original input text. For A.C., C.P.and Agg, higher values are better. For PPL, lower values are better

	C.P.	PPL	AC%
Original*	100.00	12.51	0.08
Del	97.47	363.64	**92.30**
Del&ret	97.50	242.33	71.70
B-GST	96.73	1166.4	10.10
G-GST	99.11	621.50	38.80
CAE	95.60	795.58	83.70
Our model	**99.71**	**76.75**	88.10

Table 2: **Yelp dataset-** Transfer strength and Content preservation scores for the models for the . **C.P.**: Content preservation, **PPL**: Fluency (Perplexity), **Accuracy**: Style transfer accuracy, **Original*:** refers to the original input text. . For A.C., C.P.and Agg, higher values are better. For PPL, lower values are better

	C.P.	PPL	AC%
Original*	100.00	11.39	17.80
Del	98.70	**41.03**	33.79
Del&ret	98.25	57.73	30.90
B-GST	95.94	141.81	23.90
G-GST	97.28	70.24	21.00
CAE	98.48	43.78	32.09
BST	95.49	63.33	**68.80**
Our model	**99.05**	45.17	43.20

4.3 Results and discussion

From Table 1, as we expected from the compared models, the models that perform considerably well in one metric suffer significantly in other metrics. For instance, Delete-Only (Del) produces the best transfer accuracy but lags behind other models in content preservation and fluency. For content preservation and fluency, our model produces improved results over all the other models. This result is consistent with our expectation of improving content preservation with our techniques. Again, the accuracy score (second highest) produced by our model confirms the claim that our model preserves content information without a significant drop in transfer accuracy.

From Table 2, the same observation is made for gender obfuscation; models that perform very

[5] https://www.kaggle.com/c/Jigsaw-unintended-bias-in-toxicity-classification/data

well in one metric fall short in other metrics. BST produces the best style transfer accuracy but at the same time has the worst content preservation score.

From the results from both datasets, one key observation is that models that perform very well in one metric tend to fall short in other metrics. This goes to show the difficulty for style transfer models to preserve content information whilst maintaining a strong transfer accuracy. This observation is confirmed by previous works (Li et al., 2018; Wu et al., 2019; Hu et al., 2020) which mention the general trade-off between style transfer accuracy and content preservation. Our model shows good results in maintaining a good balance across all metrics. Some text samples from our experiments are shown in Appendix A Table 4. Also, in Appendix A, Table 5 and Table 6 show the results from an ablation analysis on the Yelp dataset, where we strip off components of our model to analyze the effect. Text samples from the ablation study are also provided in Appendix A, Table 7 and Table 8.

5 Related work

He et al. (2021) propose DePen, a Detect and Perturb approach to neutralize biased texts, using graduate school admissions as a case study. Sun et al. (2021) propose a method that aims to rewrite English texts with gender-neutral English (in particular, the use of singular *they* for gender pronouns) using a combination of regular expressions, a dependency parser, and GPT-2 (Radford et al., 2019) model. Nogueira dos Santos et al. (2018) propose an RNN-based auto-encoder model to neutralize offensive language on social media, using a combination of classification loss and reconstruction loss to ensure style transfer and to improve text generation. In a different but related context, Reddy and Knight (2016b) propose a gender obfuscation technique to disguise or change the gender of an author of a text as a means of privacy protection or for the prevention of inadvertent discrimination against the author. Their method is a word substitution technique based on word2vec (Mikolov et al., 2013).

6 Conclusion

In this work, we introduce a style transfer model that can be used to mitigate bias in textual data. We show that explicit keyword replacement can be effectively combined with latent content representation to improve the content preservation of text style transfer models.

As part of our future work, we intend to expand this work to other languages, we plan to explore possible improvements to the model such as adversarial learning, and also to include human evaluators for qualitative evaluation. Again, we intend to investigate other forms of attributes beyond tokens, such as sentence length, and how that affects bias in textual data. We also plan to apply our model as a preprocessing technique to train fair language models. We believe this could significantly reduce biases found in automated language systems.

7 Ethical considerations

Works like Dev et al. (2021) have drawn attention to gender exclusivity and issues relating to non-binary representation in NLP, particularly in the English language. For practical constraints such as the limited availability of non-binary gender data and/or the significant under-representation of non-binary gender identities in available datasets, we limit this study to a binary definition of gender. For the same reasons stated above, our definition of gender is analogous to female and male definitions of sex (Walker and Cook, 1998). Although this is an obvious limitation to our work, we believe this work opens the door to extensively explore similar issues in non-binary gender settings, which need a more expansive discussion.

Since the definition of a biased text is highly domain, context, and task dependent, especially when it relates to the use of language (English in this case), our approach identifies "biased" and "neutral" texts as per how they are defined or annotated in the training data for a specific task. Hence, the labels (fair or biased) assigned to certain text examples may not be perceived accordingly in other settings and tasks. We also note that, although the use of explicit gender terms in certain domains may be deemed to introduce biases (in some recruitment scenarios for instance), this practice may be acceptable or even encouraged in other domains such as in text discussions about diversity and sexism.

Acknowledgements

This research was supported by the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. NIPS'16, page 4356–4364, Red Hook, NY, USA. Curran Associates Inc.

Jeffrey Dastin. 2018. Amazon scraps secret ai recruiting tool that showed bias against women. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G. Accessed: 2021-08-21.

Paul R Daugherty, H James Wilson, and Rumman Chowdhury. 2019. Using artificial intelligence to promote diversity. *MIT Sloan Management Review*, 60(2):1.

Sunipa Dev, Masoud Monajatipoor, Anaelia Ovalle, Arjun Subramonian, Jeff Phillips, and Kai-Wei Chang. 2021. Harms of gender exclusivity and challenges in non-binary representation in language technologies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1994, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423.

Danielle Gaucher, Justin Friesen, and Aaron C Kay. 2011. Evidence that gendered wording in job advertisements exists and sustains gender inequality. *Journal of personality and social psychology*, 101(1):109.

Zexue He, Bodhisattwa Prasad Majumder, and Julian McAuley. 2021. Detect and perturb: Neutral rewriting of biased and sensitive text via gradient-based decoding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4173–4181, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Lea Hodel, Magdalena Formanowicz, Sabine Sczesny, Jana Valdrová, and Lisa von Stockhausen. 2017. Gender-fair language in job advertisements: A cross-linguistic and cross-cultural analysis. *Journal of Cross-Cultural Psychology*, 48(3):384–401.

Zhiqiang Hu, Roy Ka-Wei Lee, and Charu C Aggarwal. 2020. Text style transfer: A review and experiment evaluation.

Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only.

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

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and*

Short Papers), pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194, Melbourne, Australia. Association for Computational Linguistics.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. 2017. Personalized machine translation: Preserving original author traits. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1074–1084, Valencia, Spain. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Sravana Reddy and Kevin Knight. 2016a. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 17–26.

Sravana Reddy and Kevin Knight. 2016b. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 17–26, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

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*, NIPS'17, page 6833–6844, Red Hook, NY, USA. Curran Associates Inc.

Nidhi Subbaraman. 2017. Scientists taught a robot language. it immediately turned racist. `https://www.buzzfeednews.com/article/nidhisubbaraman/robot-racism-through-language`. Accessed: 2021-08-22.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text style transfer.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3267–3277. Association for Computational Linguistics.

Tony Sun, Kellie Webster, Apu Shah, William Yang Wang, and Melvin Johnson. 2021. They, them, theirs: Rewriting with gender-neutral english. *arXiv preprint arXiv:2102.06788*.

Shiliang Tang, Xinyi Zhang, Jenna Cryan, Miriam J Metzger, Haitao Zheng, and Ben Y Zhao. 2017. Gender bias in the job market: A longitudinal analysis. volume 1, pages 1–19. ACM New York, NY, USA.

Phillip L Walker and Della Collins Cook. 1998. Brief communication: Gender and sex: Vive la difference. *American Journal of Physical Anthropology: The Official Publication of the American Association of Physical Anthropologists*, 106(2):255–259.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5271–5277. International Joint Conferences on Artificial Intelligence Organization.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. NIPS'18, page 7298–7309, Red Hook, NY, USA. Curran Associates Inc.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018. Style transfer as unsupervised machine translation.

A Appendix

Table 3: Dataset statistics

Dataset	Attributes	Classifier	Train	Dev	Test
Jigsaw	Sexist	24K	32K	1K	1K
	Neutral	24K	92K	3K	3K
Yelp	Male	100K	100K	1K	1K
	Female	100K	100K	1K	1K

Table 4: Sample text outputs from experiments

Gender bias mitigation (biased → neutral): Jigsaw	
input text	i hope the *man* learned his lesson to slow down and buckle up .
our model	i hope the *driver* learned his lesson to slow down and buckle up .
input text	i married a wonderful mature , loyal and dedicated foreign *women* while working abroad ...
our model	i married a wonderful mature , loyal and dedicated foreign *person* while working abroad ...
Gender obfuscation (female → male): Yelp	
input text	overall , worth the *extra* money to *stay* here .
our model	overall , worth the *damn* money to *eat* here .
input text	i had prosecco and my *boyfriend* ordered a beer .
our model	i had prosecco and my *wife* ordered a beer .

Table 5: Ablation study of our model on the **Jigsaw gender dataset**. **Without-LR**: model with soft sampling (class constraint) but no latent content representation, **Without-LR&SS**: model with no class constraint and no latent content representation

	C.P	PPL	ACC%
Our model	**99.71**	**76.75**	88.10
Without-LR	99.69	98.87	**93.44**
Without-LR&SS	99.70	98.68	93.44

Table 6: Ablation study of our model on the **Yelp dataset**. **Without-LR**: model with soft sampling (class constraint) but no latent content representation, **Without-LR&SS**: model with no class constraint and no latent content representation. Although Without-LR has a very high accuracy score, as can be seen from the example in 8, many of the Without-LR texts are unable to preserve content information

	C.P	PPL	ACC%
Our model	**99.05**	45.17	43.20
Without-LR	96.62	45.72	**84.20**
Without-LR&SS	96.89	**41.84**	41.00

Table 7: Sample text outputs from ablation study from Jigsaw dataset

Gender bias mitigation (biased → neutral): Jigsaw	
input text	if there was an article disparaging *women* as idiots there would be a protest and a parade .
our model	if there was an article disparaging *them* as idiots there would be a protest and a parade .
Without-LR	if there was an article disparaging *muslims* as idiots there would be a protest and a parade .
Without-LR&SS	if there was an article disparaging *muslims* as idiots there would be a protest and a parade .

Table 8: Sample text outputs from ablation study Yelp dataset

Gender obfuscation (female → male): Yelp	
input text	i did not buy extra insurance !
our model	i did not buy auto insurance !
Without-LR	i did not buy life insurance !
Without-LR&SS	i did not buy the pistol !