FairFlow: An Automated Approach to Model-based Counterfactual Data Augmentation For NLP

Ewoenam Kwaku Tokpo and Toon Calders

University of Antwerp, Antwerp, Belgium
{ewoenamkwaku.tokpo,toon.calders}@uantwerpen.be

Abstract. Despite the evolution of language models, they continue to portray harmful societal biases and stereotypes inadvertently learned from training data. These inherent biases often result in detrimental effects in various applications. Counterfactual Data Augmentation (CDA), which seeks to balance demographic attributes in training data, has been a widely adopted approach to mitigate bias in natural language processing. However, many existing CDA approaches rely on word substitution techniques using manually compiled word-pair dictionaries. These techniques often lead to out-of-context substitutions, resulting in potential quality issues. The advancement of model-based techniques, on the other hand, has been challenged by the need for parallel training data. Works in this area resort to manually generated parallel data that are expensive to collect and are consequently limited in scale. This paper proposes FairFlow, an automated approach to generating parallel data for training counterfactual text generator models that limits the need for human intervention. Furthermore, we show that FairFlow significantly overcomes the limitations of dictionary-based word-substitution approaches whilst maintaining good performance.

Keywords: Natural language processing · Bias mitigation · Counterfactual Data Augmentation

1 Introduction

Despite their growing popularity and unprecedented performance in various application domains, language models (LMs) continue to be plagued with issues of harmful societal biases and stereotypes that have been shown to have detrimental social effects [4]. The biggest contributing factor is the encapsulation of societal biases in everyday language, as is well-documented [1,19,12]. LMs heavily rely on such textual data, now digitalized on various online outlets, as training data, causing them to mirror these biases [25].

In Natural Language Processing (NLP), similar to many machine learning domains, bias mitigation generally occurs at three intervention avenues: the training data, the learning procedure, or the model output [15]. Since model bias traces its roots to the training data, mitigating bias at the training data level has proven very effective [10,6]. One such approach, Counterfactual Data Augmentation (CDA) [5], seeks to remove spurious correlations between attributes in the training data by evening out the distribution of words that characterize demographic attributes in the context of neutral words that should ideally not be demographically aligned. Specifically, explicit attribute-defining words are replaced with their counterfactual equivalents from complementary demographic groups for every text instance. To illustrate this with an example, an instance of "*She is a nurse*" will be augmented with "*He is a nurse*" in the case of mitigating gender bias. This follows the intuition that in an ideal dataset, the association between gender attributes and target attributes like professions will be even for different gender groups.

Key works, such as [27,16,28], introducing CDA as a bias mitigation technique adopt a word substitution approach based on dictionaries. These word substitution methods are prone to grammatical incoherence because of out-of-context substitutions and omitted word pairs. Because dictionary compilations are often incomplete [8], a direct word-substitution approach will not generalize to omitted words. Take for instance (***Bachelor*** *and* ***Masters*** *degree* v. ***Spinster*** *and* ***Mistresses*** *degree*) and (*she taught* ***herself*** v. *he taught* ***herself***) which were common issues we observed with some methods. Additionally, the dictionaries are manually compiled, which not only incurs potential costs but manually compiling counterfactual word pairs for certain demographics may be intrinsically challenging.

Although generative language models like GPT-related models [21] have surged in popularity, their adoption for CDA has been limited due to the relative unavailability of parallel data needed for training. As such, model-based solutions resort to manually compiling parallel training data, a process that is both costly and constrained. This challenge is exacerbated by the fact that training models on limited parallel data can impair performance [29]. Although large conversational models like ChatGPT generate good counterfactuals in a zero-shot setting, they are not efficient in low-resource environments. In this work, we focus on low-resource/resource-efficient techniques that can be deployed in low-resource environments.

The primary contribution of this paper is to explore an automated approach to generate parallel training data for a given demographic axis that requires minimal human intervention. Our approach takes from a user a prompt – in the form of a single word-pair – that describes a demographic axis. This pair is subsequently used to model a demographic subspace from which other words that define the demographic attribute can be sampled from a given corpus of text. Using an invertible flow-based model [9], counterfactual words are generated for sampled words. Thereafter, an error correction approach is used in tandem with direct word substitution to generate parallel data to fine-tune a generative language model to generate counterfactual texts. We call our approach and the resultant counterfactual text generation model *FairFlow*. This entire process is simply depicted in a four-step process in Fig. 1. As opposed to existing works, which will be discussed in Section 2, FairFlow does not rely on human-generated parallel data for training and eliminates the need for manually compiled word-pair dictionaries.

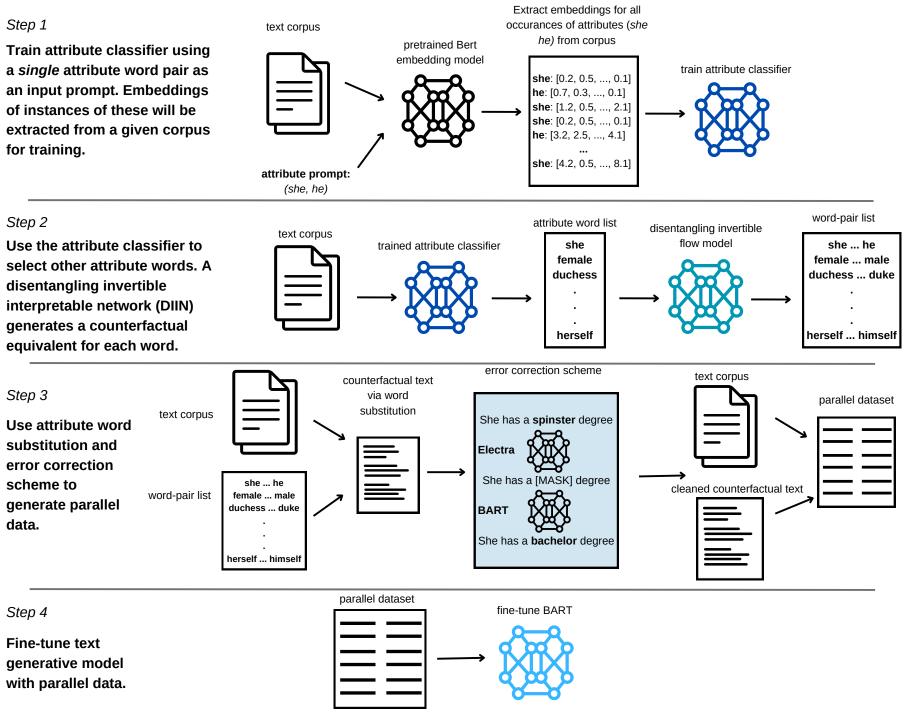

Step 1
Train attribute classifier using a *single* attribute word pair as an input prompt. Embeddings of instances of these will be extracted from a given corpus for training.
text corpus
attribute prompt: (she, he)
pretrained Bert embedding model
Extract embeddings for all occurances of attributes (she he) from corpus
she: [0.2, 0.5, ..., 0.1]
he: [0.7, 0.3, ..., 0.1]
she: [1.2, 0.5, ..., 2.1]
she: [0.2, 0.5, ..., 0.1]
he: [3.2, 2.5, ..., 4.1]
...
she: [4.2, 0.5, ..., 8.1]
train attribute classifier
Step 2
Use the attribute classifier to select other attribute words. A disentangling invertible interpretable network (DIIN) generates a counterfactual equivalent for each word.
text corpus
trained attribute classifier
attribute word list
she
female
duchess
herself
disentangling invertible flow model
word-pair list
she ... he
female ... male
duchess ... duke
herself ... himself
Step 3
Use attribute word substitution and error correction scheme to generate parallel data.
text corpus
word-pair list
she ... he
female ... male
duchess ... duke
herself ... himself
counterfactual text via word substitution
error correction scheme
She has a spinster degree
Electra
She has a [MASK] degree
BART
She has a bachelor degree
text corpus
cleaned counterfactual text
parallel dataset
Step 4
Fine-tune text generative model with parallel data.
parallel dataset
fine-tune BART

Fig. 1. An end-to-end description of Fairflow, described in four steps: 1) train a classifier to identify attribute words from a corpus; 2) generate counterfactual equivalents for attribute words using an invertible generative flow model; 3) use a word substitution scheme and our proposed error-correction scheme to make the parallel text more fluent and realistic; 4) fine-tune a generative model with the generated parallel data.

In summary, this paper explores and proposes techniques to develop a robust model-based counterfactual generator in the absence of parallel training data. Key contributions include:

1. An automated approach to compiling dictionaries of word pairs that only requires a user to input a word-pair prompt that describes a demographic axis.
2. We proposed an error correction approach to generate parallel data from dictionary word substitutions.
3. We train a counterfactual model using our generated parallel data and show that the error correction approach not only improves the grammatical composition of the model but also improves the generalization of the model.

We make our implementation code and materials for FairFlow available[1].

2 Background and Related Literature

Early works on CDA used simple rule-based word-substitution approaches for counterfactual data augmentation. Specifically, they created dictionaries of attribute word pairs and used matching rules to swap words [6]. Later works began to incorporate grammatical information like part-of-speech tags to swap attribute words [27]. In the absence of interventions for named entities, Lu et al. [16] do not augment sentences or text instances containing proper nouns, and named entities as generating counterfactuals without proper name interventions could result in semantically incorrect sentences. Zhao et al. [27] circumvented this by anonymizing named entities by replacing them with special tokens. Lamenting on the aforementioned lack of parallel corpus for training neural models, Zmigrod et al. [28] used a series of unsupervised techniques such as dependency trees, lemmata, part-of-speech tags, and morpho-syntactic tags for counterfactual generation. Hall-Maudsley et al. [17] improve on Zmigrod et al. by incorporating a names intervention method to resolve the challenges of generating counterfactuals for named entities. They achieve this using a bipartite graph to match first names.

Because the aforementioned techniques rely on dictionary word replacement techniques and ignore the context of the text, they are prone to generating ungrammatical texts. Additionally, the inability of these techniques to resolve out-of-dictionary words not only preserves certain attribute correlations but also introduces errors. We illustrate two instances of such limitations using the word substitution approach by Hall-Maudsley et al. on the Bias-in-bios dataset [6]; 1) *"Memory received her **Bachelor** and **Masters** of Accountancy..."* produces *"Memory received his **Spinster** and **Mistresses** of Accountancy..."* due to the polysemous nature of *bachelor* and *master*; 2) *"Laura discovered her passion for programming after teaching **herself** some Python..."*, is transformed into *"Anthony discovered his passion for programming after teaching **herself** some Python..."* as the gender pronouns *herself* and *himself* are excluded from the dictionary compiled by Hall-Maudsley et al.

More recently, sequence-to-sequence model-based approaches to counterfactual generation have been proposed [26,20]. Wu et al. [26] propose Polyjuice, a generative counterfactual model for diverse use cases like counterfactual explanations. They generate parallel data by pairing naturally occurring sentences in a corpus based on edit distances. Although effective for explanations, such an approach is not applicable for bias mitigation as attribute words, in the case of the latter, have to be specifically defined and replaced. Specifically for bias mitigation, Qian et al. [20] introduce the *perturber*, which is a Bart[14] model fine-tuned on a human-generated parallel text. However, their approach only generates counterfactuals for specific user-defined entities in a text. eg. *original:"Torii chose to remain behind, pledging that he and his **men** would fight..."*,

[1] https://github.com/EwoeT/FairFlow

rewrite:"Tara chose to remain behind, pledging that she and her ***men*** *would fight ...".* As earlier stated, such manually compiled datasets are expensive and are only available on small scales, which can degrade performance [29]. Additionally, similar manual efforts must be solicited for every language domain for which counterfactuals have to be generated. As opposed to existing works, the main advantage of our work is the non-reliance on human-generated parallel data and word lists.

3 Approach

Our entire approach can be summarized in four steps as illustrated in Fig. 1. The process commences with training a classifier to detect attribute words in a corpus, after which counterfactuals for these attribute words are generated using an invertible flow model. Parallel data is thereafter created by using a combination of word substitution and an error-correction scheme. Finally, a generative model is fine-tuned using the generated parallel data. We expound on these steps in the following subsections.

3.1 Attribute classifier training

To select a list of words that characterize a given demographic axis, e.g. gender, we first train an attribute classifier that approximates the attribute subspace. To do this, the user first inputs a prompt in the form of a single pair of words that describes a given demographic axis, e.g., (she, he) in the case of gender. Using a pretrained contextualized word embedding model, contextualized word representations are generated for each appearance of the input words within a given text corpus — we take *BERT-base-uncased* [7] as our choice of representation model. These embeddings are used to train a classifier to approximate the demographic subspace. Formally, consider the word-pair (x_a, x_b) that define a demographic axis, we obtain two sets $Z_a = \{z_{a_1}, z_{a_2}, ..., z_{a_n}\}$ and $Z_b = \{z_{b_1}, z_{b_2}, ..., z_{b_n}\}$ where $z_{a_i} \in R^d$ and $z_{b_i} \in R^d$ are context-specific vector representations of instances of x_a, x_b respectively, generated from a text corpus V by a pretrained embedding model E; so that $E(x_i, c_i) = z_i$ if x_i is an instance of a word x and c_i is its context. We estimate the demographic subspace by training a classifier H to maximizing the objective $\sum\limits_{z_i \in \{Z_a \cup Z_b\}} log(P(y|z_i))$, where $y = \{a, b\}$ is the class label of z_i. H is parameterized as a feed-forward neural network with one hidden layer and Gelu non-linear activation.

3.2 Generating word-pair list

Selecting attribute words. Given a demographic subspace, we select all words that lie within the attribute-defining regions of the subspace. This process is formally described as follows. Given our initial corpus V, we select words $x_i \in V$ based on the criterion $P(y|E(x_i, c_i); \Theta_H) > \phi$ where Θ_H represents the parameters that define H and ϕ is a predefined threshold. Z_a is thus expanded to include all words that have at least an instance satisfying $P(y = a|E(x_i, c_i); \Theta_H) > \phi$ and Z_b to include all words with at least an instance satisfying $P(y = b|E(x_i, c_i); \Theta_H) > \phi$. Although some neutral words may be included in these sets, they do not produce any counterfactual equivalent in the next stage, hence making no difference.

Generating counterfactual word-pairs with DIIN. The first step in generating counterfactual equivalents for the set of words Z_a and Z_b is to define a transformation T from the original embedding space into an "interpretable" space where an embedding is factorizable into independent components. We train T to constrain attribute information *only* to the first k dimensions (we will collectively refer to these dimensions as K) of a word in the interpretable space. By so doing, K can be swapped to alter the attribute (eg. gender) of the word. We implement T using a flow-based generative model [13,18,9]; specifically, we use the disentangling invertible interpretation network (DIIN) architecture by Esser et al. [11].

Formally, given the contextualized representation z of a word x, the goal is to learn a transformation T that maps the original representation $z \in R^d$ to an interpretable representation $\tilde{z} \in R^d$ s.t. $T(z) = \tilde{z}$. The interpretable representation $\tilde{z}$ is sampled from a base distribution $\tilde{z} \sim p_{\tilde{\mathcal{Z}}}(\tilde{z})$ – a standard Gaussian distribution in this case. Using the change of variable theorem, T is learned by maximizing the log-likelihood

$$\log(p_{\mathcal{Z}}(z)) = \log(p_{\tilde{\mathcal{Z}}}(T(z))) + \log(|\det(\frac{\partial T(z)}{\partial z})|) \tag{1}$$

To constrain attribute information only to K, we pair embeddings of words that have the same attribute F and train T to generate similar values for both embeddings in their first k dimensions in the interpretable space. Mathematically, Given a pair of embeddings (z_{a_1}, z_{a_2}) that belong to the same demographic group such that $F_{z_{a_1}} = F_{z_{a_2}}$, the objective is achieved by minimizing the loss function:

$$\begin{aligned}\mathcal{L}(z_{a_1}, z_{a_1}|F) = ||T(z_{a_1})_D||^2 - \log(\det(T(z_{a_1})))\\ +||T(z_{a_2})_{(D\setminus K)}||^2 - \log(\det(T(z_{a_2})))\\ +\frac{||T(z_{a_2})_K - \sigma T(z_{a_1})_K||^2}{1-\sigma^2}\end{aligned} \tag{2}$$

where D is a term to collectively refer to all d components of the embedding. $\sigma \in (0, 1)$ is a positive correlation factor that determines the strength of the correlation between $z_{a_{2K}}$ and $z_{a_{1K}}$. We also use the dimensionality estimation approach of Esser et al. to estimate the dimensionality of K.

Once our invertible flow model has been trained to constrain F to the first k dimensions of $\tilde{z}$ (in the interpretable space), we replace $z_{a_{iK}}$ which is the first k dimensions of $\tilde{z}_{a_i}$ with K'_b; such that $z_{a_{iK}} \rightarrow K'_b$, where $K'_b = \frac{1}{N}\sum_{i=0}^{N} z_{b_{iK}}$ is the

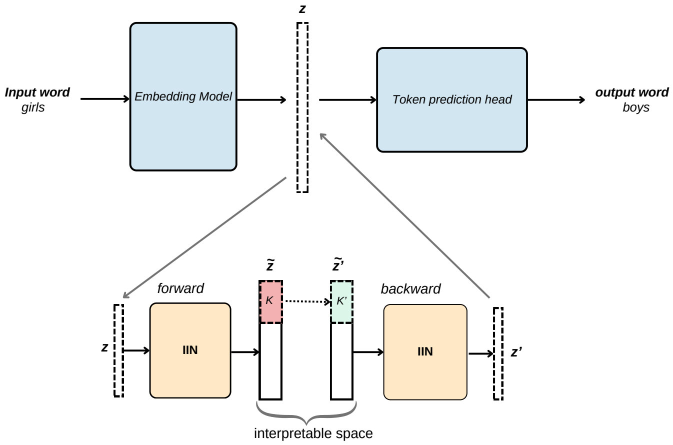


Fig. 2. Counterfactual word generation using an invertible interpretation flow network IIN.

average of the first k dimensions of the complementary demographic group. This process is depicted in Fig. 2. We use a majority voting scheme to then select the most frequent equivalent generated for each word. An output example of this process obtained using a {*"she"*, *"he"*} prompt is shown in Fig. 3. We then extend this list using the names intervention approach of Hall-Maudsley et al. to generate counterfactuals for names.

3.3 Error correction

With the word pairs generated from the previous phase, we use the word substitution approach of Hall-Maudsley et al. to build a base corpus. To transform this base corpus into fluent and realistic text labels for our parallel training data, we proposed an error correction scheme which we describe below in two steps.

Erratic token detection. The idea here is to detect and mask tokens that have a low probability of appearing in the context of a given text; following $t_i = t_{<mask>}$ if $P(t_i|T \setminus t_i) < \theta$, where T is the sequence of tokens, t_i is the ith token in T, and θ is a predefined threshold value. We define the resulting masked text as T_{Π}. This is achieved using a pretrained Electra model [3]. Electra is an LM pretrained using a text corruption scheme – text instances are corrupted by randomly replacing a number of tokens with plausible alternatives from BERT. Electra is then trained to predict which tokens are real and fictitious.

Since the use of wordpiece tokenization causes issues (as a word can be broken down into multiple subtokens) if a subtoken is selected for masking, we replace the entire sequence of associated subtokens with a `<mask>` token. For instance, *"The men are duchesses"*, in a wordpiece tokenization could be decomposed to

actress -- actor
alice -- edward
aunt -- uncle
barbara -- david
baroness -- baron
beautiful -- handsome
countess -- count
daughter -- son
daughters -- sons
elizabeth -- john
empress -- emperor
female -- male
females -- males
feminine -- masculine
girl -- boy
girlfriend -- boyfriend
girls -- boys
granddaughter -- grandson
grandmother -- grandfather
heiress -- heir
her -- his

hers -- his
herself -- himself
jane -- john
jess -- matt
ladies -- gentleman
lady -- lord
mary -- john
miss -- mr
mom -- dad
mother -- father
mothers -- fathers
mrs -- mr
ms -- mr
niece -- nephew
princess -- king
queen -- king
queens -- kings
sarah -- michael
she -- he
sister -- brother

sisters -- brothers
soprano -- tenor
sorority -- fraternity
teresa -- luis
virginia -- william
widow -- man
wife -- husband
woman -- man
women -- man
actor -- actress
ap -- her
baron -- lady
boy -- girl
boyfriend -- girlfriend
boys -- girls
brother -- sister
brothers -- sisters
christopher -- elizabeth
count -- countess
dad -- mom
david -- barbara

earl -- countess
edward -- alice
emperor -- empress
emperors -- empress
father -- mother
fathers -- mothers
fraternity -- sorority
gentleman -- ladies
grandfather -- grandmother
grandson -- granddaughter
guy -- girl
handsome -- beautiful
he -- she
heir -- heiress
henry -- elizabeth
him -- her
himself -- herself
his -- her
human -- female
husband -- wife
jesus -- mary
john -- jane

king -- queen
kings -- queens
lord -- lady
luis -- teresa
male -- female
males -- females
man -- woman
masculine -- feminine
matt -- jess
maximilian -- mary
michael -- sarah
mr -- mrs
nephew -- niece
peter -- mary
richard -- elizabeth
robert -- mary
sir -- lady
son -- daughter
sons -- daughters
tenor -- soprano
thomas -- elizabeth
uncle -- aunt
william -- virginia

Fig. 3. An autmatically compiled dictionary using the input prompt {*"she"*, *"he"*}. Words are discovered using the attribute classifier, and the counterfactuals are generated using the disentangling invertible interpretation network.

["The", "men", "are", "duchess", "##es"], Consequently, when *"duchess"* is identified as an erratic token, the masking scheme replaces the entire subsequence *["duchess", "##es"]*, thereby, generating *"The men are* `<mask>`*"*.

Text insertion with BART. Having obtained our masked intermediary texts, we generate plausible token replacements for each masked token. Since a `<mask>` token could correspond to multiple subword tokens, the replacement generator should be capable of generating multiple tokens for a single `<mask>` instance, making it suitable to use a generative model – pretrained BART [14] – to predict these replacement tokens. Because Masked Language Modeling is one of BART's pretraining objectives, we can utilize it in its pretrained form without the need for finetuning. Given T_{Π} from the previous step, the BART model tries to predict the correct infilling x using the context of T_{Π}.

3.4 Training the generative model

The final stage of the approach is to fine-tune a BART model using the parallel data obtained from the previous steps. The BART generator takes the original text as input and is trained to autoregressively generate the counterfactual of the source text using the corresponding parallel counterfactual texts as labels in a teacher-forcing manner [24]. We formulate this as:

$$\mathcal{L}_{generator} = -\sum_{t=1}^{k} logP(y_t|Y_{<t}, X) \tag{3}$$

Where X and Y are the source and target texts, respectively, $y_t \in Y$ is the t^{th} token in the target text, and $Y_{<t}$ refers to all tokens in Y preceding y_t.

4 Experimental set-up

This section describes key implementation details of our work and the evaluation framework. We specifically evaluate gender bias in the binary sense within the English language domain.

4.1 Training set-up

The main corpus for training the attribute classifier and the disentangling invertible flow model comprises Wikipedia articles via Wikimedia dumps[2].

4.2 Evaluation datasets

For the appraisal of our model, we used the datasets discussed below. These datasets, upon which various CDA interventions were applied, were used to train a classification model on a downstream task. These datasets were only used for evaluation purposes and were not included in training Fairflow.

1. **Bias-in-bios**: This dataset provided by De-Arteaga et al. [6] contains Wikipedia profiles of professionals. The dataset originally contained labels corresponding to 28 distinct professions alongside the gender labels of the profiled individuals. We reclassified the professions into binary labels, aligning them with male-dominated and female-dominated occupations according to gender distribution. This categorization was done for two reasons. The first was to simplify the classification task from multiclass to binary. Secondly, this enabled us to easily induce bias by creating an imbalance between gender and class labels.
2. **ECHR**: The ECHR dataset by Chalkidis et al. [3] [2] contains case facts from the European Court of Human Rights (ECHR) on human rights breaches by European states. It further contains information on the gender of the applicant, human rights articles that were violated, and the defendant state (Central-Eastern European states *v.* all other states). The primary classification task here was to predict the defendant's state based on the case facts.
3. **Jigsaw**: This dataset[4] contains public comments from the now defunct online platform Civil Comments. The primary classification task for this dataset was toxicity detection.

[2] https://dumps.wikimedia.org

[3] https://huggingface.co/datasets/coastalcph/fairlex

[4] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

For all the evaluation datasets, we maintained a balanced gender and class label distribution in the test sets as shown in Table 1. The training sets for the Bias-in-bios and the Jigsaw datasets were sampled with an imbalance to induce bias following the observations of Dixon et al. [10]. The training set for ECHR was left relatively balanced with the additional purpose of providing a baseline.

		Train			Test		
Dataset	**Task**	**Number** (K)	**Positive class** %	**Females in Pos.** %	**Number** (K)	**Positive class** %	**Females in Pos.** %
Bias-in-bios	Career	18	50	12	4	50	50
ECHR	State	7	18	41	1	50	50
Jigsaw	Toxicity	5	47	77	1	50	50

Table 1. Evaluation dataset statistics: The test sets are balanced with regard to gender and labels.

4.3 Comparative techniques

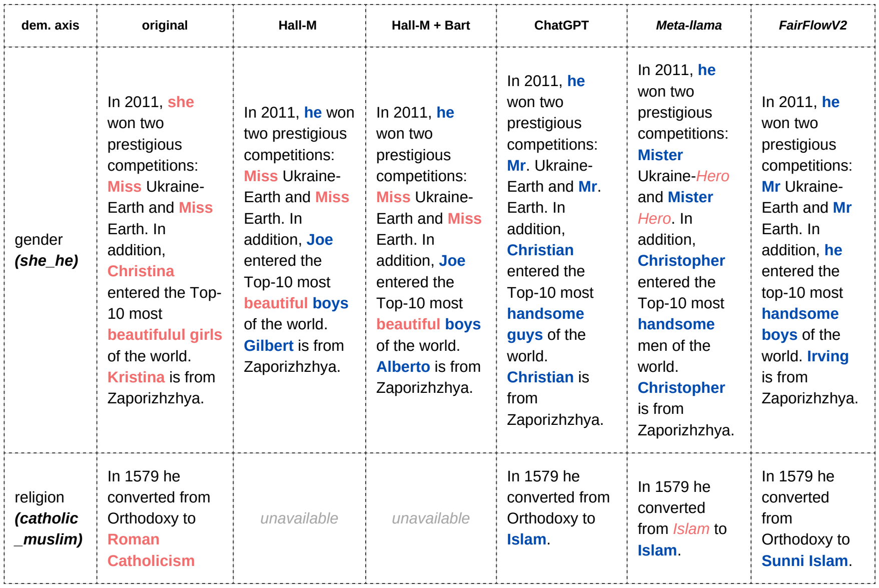
We implemented two variants of FairFLow: *FairFLowV1* and *FairFLowV2*, and compared them to three CDA setups. 1) *original* is the unaugmented original text; 2) *Hall-M* uses the direct word-substituion approach proposed by Hall-Maudsley et al [17]; 3) *Hall-M + BART* is a BART model fine-tuned with counterfactuals generated by Hall-Maudsley et al.; 4) *FairFlowv1* is a BART model fine-tuned with our error correction scheme applied to counterfactuals from Hall-Maudsley et al.; it follows the same approach of FairLow in Fig. 1 but with a manually compiled dictionary. 5) *FairFlowv2* is a BART-model fine-tuned with our full approach in Fig. 1. We take *Hall-M* and *Hall-M + BART* as our baseline approaches. We excluded *perturber* by Qian et al. [20] from our evaluation since the objective of their approach significantly differs from ours; as elaborated in Section 2.

5 Evaluation and results

We quantitatively evaluated our approach using three main criteria: *utility*, *extrinsic bias mitigation*, and *task performance*.

5.1 Utility

By utility, we refer to how realistic and effective the generated counterfactuals are by computing their fluency (perplexity) and gender transfer accuracy.



dem. axis	**original**	**Hall-M**	**Hall-M + Bart**	**ChatGPT**	***Meta-llama***	***FairFlowV2***
gender ***(she_he)***	In 2011, **she** won two prestigious competitions: **Miss** Ukraine-Earth and **Miss** Earth. In addition, **Christina** entered the Top-10 most **beautiful girls** of the world. **Kristina** is from Zaporizhzhya.	In 2011, **he** won two prestigious competitions: **Miss** Ukraine-Earth and **Miss** Earth. In addition, **Joe** entered the Top-10 most **beautiful boys** of the world. **Gilbert** is from Zaporizhzhya.	In 2011, **he** won two prestigious competitions: **Miss** Ukraine-Earth and **Miss** Earth. In addition, **Joe** entered the Top-10 most **beautiful boys** of the world. **Alberto** is from Zaporizhzhya.	In 2011, **he** won two prestigious competitions: **Mr**. Ukraine-Earth and **Mr**. Earth. In addition, **Christian** entered the Top-10 most **handsome guys** of the world. **Christian** is from Zaporizhzhya.	In 2011, **he** won two prestigious competitions: **Mister** Ukraine-*Hero* and **Mister** *Hero*. In addition, **Christopher** entered the Top-10 most **handsome** men of the world. **Christopher** is from Zaporizhzhya.	In 2011, **he** won two prestigious competitions: **Mr** Ukraine-Earth and **Mr** Earth. In addition, **he** entered the top-10 most **handsome boys** of the world. **Irving** is from Zaporizhzhya.
religion ***(catholic _muslim)***	In 1579 he converted from Orthodoxy to **Roman Catholicism**	*unavailable*	*unavailable*	In 1579 he converted from Orthodoxy to **Islam**.	In 1579 he converted from *Islam* to **Islam**.	In 1579 he converted from Orthodoxy to **Sunni Islam**.

Fig. 4. Text samples from Bias-in-bios and Wikipedia demonstrate that *FairFlow* and *ChatGPT-4* generate more robust counterfactual texts. Compared to *ChatGPT-4*, *Meta-llama-3-8B-Instruct* generates more inaccurate counterfactuals.

Grammatical correctness and fluency We used a referenceless fluency metric due to the relative unavailability of parallel data. As we noted earlier, the parallel data used by Qian et al. only contains counterfactuals for only specific user-defined entities and is thus not suitable for evaluating our work. Similar to Wu et al. [26], we score fluency by computing the perplexity of the generated text using pretrained GPT-2 [22]. A low perplexity implies that a given text conforms well to the probabilistic distribution of natural text as learned by the pretrained language model.

Based on our earlier assertion about how out-of-context substitutions impair fluency, our error correction approach should expectedly increase fluency (reduce perplexity). We confirm this in Table 2 as we see that fluency is consistently improved in both *FairFlowV1* and *FairFlowV2*.

Transfer accuracy Here, similar to Tokpo et al. [23], we computed the percentage of texts that were converted from the source attribute to the target attribute, i.e., female to male or vice versa. We fine-tuned a BERT model to predict the gender of the text. We quantified gender transfer accuracy as $1 - probability\_of\_original\_attribute$. We expect the original text to have a very low transfer accuracy, as its attributes would remain the same. As shown in Table 2, FairFlowV2 especially shows strong fluency scores whilst maintaining

Approach	*PPL* ↓			*Transfer Accuracy* ↑		
	Bios	**Jigsaw**	**ECHR**	**Bios**	**Jigsaw**	**ECHR**
*Original**	41.023	69.67	32.88	0.04	15.96	36.14
Hall-M	43.51	76.37	33.70	98.60	**79.00**	75.10
Hall-M + BART	47.59	83.76	39.93	98.70	78.50	71.10
FairFlowV1	42.77	65.80	33.70	**98.91**	77.99	74.69
FairFlowV2	**39.86**	**63.99**	**33.33**	98.51	70.736	**76.51**

Table 2. PPL (*left*) of generated text using various CDA techniques. Lower scores indicate better fluency. Gender transfer accuracy (*right*) of the various CDA interventions. This indicates the percentage of counterfactual instances that were correctly resolved to new gender styles. *The original samples have very low accuracies because original gender is preserved.*

a good transfer accuracy. This shows that automating the dictionary generation process does not materially impair transfer accuracy.

5.2 Extrinsic bias mitigation

We trained a BERT classifier using the downstream classification tasks corresponding to the respective datasets and computed the *True Positive rate difference (TPRD)* and *False Positive rate difference (FPRD)* between two gender groups as in the case of De-Arteaga et al. [6]. $TPRD = P(\hat{y} = 1|y = 1, A = a) - P(\hat{y} = 1|y = 1, A = a')$ and $FPRD = P(\hat{y} = 1|y = 0, A = a) - P(\hat{y} = 1|y = 0, A = a')$. Where y is the true label, $\hat{y}$ is the predicted label, and A is the gender group variable.

We show in Table 3 consistently high TPRD scores for FairFlow1; this further buttresses the evidence that our approach to error correction works effectively and enhances bias mitigation whilst improving fluency. Similar to our findings for transfer accuracy, we find that automating dictionary compilation does not compromise bias mitigation much, as FaiFlowV2 maintains a good mitigating effect.

5.3 Task performance

We carried out the task performance test to observe the extent to which bias mitigation impacts the task model's performance. Because we maintain a balanced distribution for our test sets, we expect the fairer models to have better performance. Specifically, we computed the accuracy and F1 scores for the default classification task of the respective datasets. In Table 4, FairFlow1 shows the most improved performance in general, particularly in accuracy. We again show from the strong performance of FairFlowV2, how effective an automatically generated dictionary could be.

Approach	*TPRD* ↓			*FPRD* ↓		
	Bios	**Jigsaw**	**ECHR**	**Bios**	**Jigsaw**	**ECHR**
*Original**	0.133	0.120	0.000	0.151	0.160	0.0
Hall-M	0.055	0.010	0.030	0.071	0.070	0.0
Hall-M + BART	0.051	0.025	0.010	0.074	**0.060**	0.0
FairFlowV1	**0.044**	**0.005**	**0.000**	**0.065**	0.065	0.0
FairFlowV2	0.057	0.040	0.010	0.070	0.080	0.0

Table 3. Extrinsic fairness: TPRD – True positive rate difference between male and female text instances. FPRD – False positive rate difference between male and female text instances.

Approach	ACC ↑			F1 ↑		
	Bios	**Jigsaw**	**ECHR**	**Bios**	**Jigsaw**	**ECHR**
*Original**	91.20	88.50	97.60	47.92	48.95	52.86
Hall-M	92.53	90.25	97.83	48.38	48.62	52.98
Hall-M + BART	92.64	90.62	97.36	**48.48**	49.49	52.73
FairFlowV1	**92.97**	**90.75**	98.08	48.32	**49.69**	53.11
FairFlowV2	92.81	90.00	**98.32**	48.36	49.21	**53.24**

Table 4. Task performance: Accuracy and F1 scores of classification tasks. FairFLow1 shows better performance scores in general. FairFlow2 maintains a significant bias mitigating effect despite an automated dictionary approach.

5.4 Qualitative analysis and key observations

By analyzing samples from FairFlow, ChatGPT, and the comparative models, we find that FairFLow and ChatGPT have the most grammatically coherent counterfactuals. Additionally, we find that:

1. ***Automating the dictionary compilation process does not materially impair counterfactual generation.*** As shown in Fig. 4, even with a dictionary that was automatically compiled, FirFlowV2 generates fluent and plausible counterfactuals. This is aided by the combination of the error correction scheme, which makes it more robust to grammatical errors and helps it generalize better.
2. ***A model fine-tuned on erroneous data mimics those errors.*** We observe that the error correction approach incorporated in FairFlow makes the model more robust, fluent, and grammatically coherent. The direct word replacement technique (*Hall-M*) is unable to replace out-of-dictionary words. The output of *Hall-M + BART* mirrors the same errors as *Hall-M*, showing that a generative model fine-tuned on erroneous data will mimic those errors.
3. ***ChatGPT generates good counterfactuals but has practical limitations.*** We observe that, in general, ChatGPT generates good counterfactuals in zero-shot settings but is inefficient at generating counterfactuals on a large

scale in low-resource environments. It is more costly to deploy in terms of access and infrastructural demands. Secondly, ChatGPT shows inconsistencies in generating counterfactuals for names, as it tends to skip some names for which counterfactuals could have been generated. This is more so if the names refer to public figures, which occasionally leads to grammatical incoherent outputs. This can, however, be addressed by adapting the input prompts and improving instructions through few-shot examples that intuitively describe the setting. The manner in which ChatGPT handles names can also be advantageous because it may preserve factuality of the text better, which may be a more desirable attribute in certain contexts. We also observed some irregular counterfactuals from *Meta-llama-3-8B-Instruct* in a zero-shot setting, as shown in Fig. 4. Some of the counterfactuals it generated impacted the original context of the text, which should have been retained.

6 Conclusion

In this paper, we highlight some issues that pertain to dictionary-based word-substitution counterfactual data augmentation techniques. We discuss how these techniques, relying on manually compiled dictionaries, are prone to grammatical incoherence and lack generalization outside dictionary terms. We discuss how a model-based approach is primarily inhibited by the relative unavailability of parallel corpora for training. In light of this: 1) we propose an automated dictionary generation approach that can automatically extract and generate word-pairs from a corpus with little human intervention; 2) we propose an error correction approach that can be used to generate fluent and grammatically coherent parallel text to train a generative model for CDA; 3) we combine these approaches to fine-tune a BART model for the purpose of generating counterfactual texts (we call the resulting model *FiarFLow*); 4) we show that our error correction approach significantly improves the fine-tuned model's fluency and bias-mitigating effect; 5) we also show that automating the dictionary compilation process comes at little cost to the performance of the CDA model and is a viable solution in settings where human intervention is challenging.

Limitations

The primary limitation of our work is the lack of exploration into more diverse demographic and language domains. The work mostly focuses on (binary) gender bias in English, which is a significant limitation, considering how nuanced gender can be in other languages. Due to the relative unavailability of CDA test resources in other demographic domains, such as race, the scope of evaluation in these areas is limited. Our future work will be directed towards addressing these research directions.

Another limitation of this work is its reliance on the tokenization scheme used by the embedding model, which means that words expressed in multiple subtokens are not included in the automatic compilation of the dictionary.

Ethics Statement

From an ethical perspective, the primary point to keep in mind regarding the use of counterfactual models is their impact on factuality. Since CDA approaches are designed to be *counterfactual*, they should be used cautiously in sensitive domains where factuality is essential. Secondly, CDA bias mitigation techniques like FairFlow do not automatically guarantee fairness; hence, they must be used with that understanding.

Acknowledgements

Ewoenam Kwaku Tokpo received funding from the Flemish Government under the "Vlaams AI-Onderzoeksprogramma" (Flanders AI Research Program). We also thank Marco Favier for sharing his insights and engaging in valuable discussions.

References

1. Beukeboom, C.J., Burgers, C.: Linguistic bias. In: Oxford Encyclopedia of Communication, pp. 1–19. Oxford University Press (2017)
2. Chalkidis, I., Passini, T., Zhang, S., Tomada, L., Schwemer, S.F., Søgaard, A.: Fairlex: A multilingual benchmark for evaluating fairness in legal text processing. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Dublin, Ireland (2022)
3. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
4. Dastin, J.: Amazon scraps secret ai recruiting tool that showed bias against women. In: Ethics of data and analytics, pp. 296–299. Auerbach Publications (2022)
5. Datta, A.: Gender bias in neural natural language processing. Logic, Language, and Security p. 189
6. De-Arteaga, M., Romanov, A., Wallach, H., Chayes, J., Borgs, C., Chouldechova, A., Geyik, S., Kenthapadi, K., Kalai, A.T.: 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. pp. 120–128 (2019)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Dinan, E., Fan, A., Williams, A., Urbanek, J., Kiela, D., Weston, J.: Queens are powerful too: Mitigating gender bias in dialogue generation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 8173–8188 (2020)