FuzzE: Fuzzy Fairness Evaluation of Offensive Language Classifiers on African-American English

Anthony Rios
Department of Information Systems and Cyber Security
University of Texas at San Antonio
anthony.rios@utsa.edu

Abstract

Hate speech and offensive language are rampant on social media. Machine learning has provided a way to moderate foul language at scale. However, much of the current research focuses on overall performance. Models may perform poorly on text written in a minority dialectal language. For instance, a hate speech classifier may produce more false positives on tweets written in African-American Vernacular English (AAVE). To measure these problems, we need text written in both AAVE and Standard American English (SAE). Unfortunately, it is challenging to curate data for all linguistic styles in a timely manner—especially when we are constrained to specific problems, social media platforms, or by limited resources. In this paper, we answer the question, "How can we evaluate the performance of classifiers across minority dialectal languages when they are not present within a particular dataset?" Specifically, we propose an automated fairness fuzzing tool called FuzzE to quantify the fairness of text classifiers applied to AAVE text using a dataset that only contains text written in SAE. Overall, we find that the fairness estimates returned by our technique moderately correlates with the use of real ground-truth AAVE text. **Warning**: Offensive language is displayed in this manuscript.

1 Introduction

Offensive language and hate speech pose a significant problem on social media. The use of human moderators does not scale to large online communities (e.g., Twitter). Furthermore, human moderators may write offensive language themselves, thereby corrupting the system. Recent research efforts have focused on annotation theory for offensive language and on developing better classification methods (Davidson et al. 2017; Zampieri et al. 2019). Unfortunately, as companies put offensive language classifiers into production, they may be biased against certain minority groups or linguistic styles (Sap et al. 2019). Yet, fairness is rarely evaluated before putting systems into production for a multitude of reasons. For example, a company or research group may not have the resources to collect data from all demographics of interest, or worse, data for certain groups may simply be unavailable or limited for particular topics or social platforms.

Many metrics and strategies have been proposed to evaluate fairness in recent years (Zliobaite 2015; Hardt et al. 2016; Dixon et al. 2018; Mitchell et al. 2019). Most methodologies require ground-truth demographic or linguistic style, annotations (Park, Shin, and Fung 2018; Badjatiya, Gupta, and Varma 2019). In the absence of annotated demographic data, Dixon et al. (2018) propose fuzzing methods to estimate fairness. Fuzzing has traditionally been used in software testing to find bugs or security vulnerabilities (Bird and Munoz 1983). To apply fuzzing to fairness testing, simulated data is used to analyze how predictions change if the topic of the tweet stays the same, but the text is slightly altered. For example, fuzzing techniques will randomly change demographic words (e.g., "He", "She", "husband", or "wife") in a tweet without changing its meaning. If the model's prediction changes by these modifications, then we assume the model is biased.

Typically, fuzzing techniques for software testing use blackbox methodologies (Zalewski 2015; Liu et al. 2019). Yet, the recent fuzzing approaches for fairness relies on manually created templates. Furthermore, current lexicon-based fuzzing methods are limited to single lexical items (e.g., "he" and "she"); complex syntactic constructions are ignored (e.g., "O-be-V"). Likewise, the manual curation process may not capture differences in vocabulary across all minority dialects. For example, "en" in Spanish translates to both "in" and "on" in English. Therefore, Hispanic users may say "Put the soup *on* the bowl", rather than saying "Put the soup *in* the bowl". If changing the word "on" to "in" changes the prediction of an advanced classifier, then the model is biased. Without expert domain knowledge, creating fuzzing test cases that capture subtle differences in the use of prepositions may not be obvious compared to words directly related to a specific demographic factor (e.g., "him", "her", "hispanic"). Similarly, differences in the use of language between groups may change over time, further increasing the difficulty of manual curation.

In this work, we investigate the use of style transfer to rank classification models with regard to standardized fairness metrics when minority linguistic styles are miss-

ing from the dataset. Intuitively, if a tweet is written in Standard American English (SAE), we want to answer the counterfactual-like question, "What would our model predict if this tweet was written in AAVE?" This task is important because depending on the application, sampling process, and data source, the text generated by specific minority linguistic styles may not be adequately represented in a dataset. Yet, it is important to understand how the model will perform for these groups. Therefore, this line of research can help practitioners ethically adopt machine learning methodologies without dramatically increasing data annotation and collection costs. Essentially, we hope to reduce the burden of evaluating fairness.

Our contributions are summarized below:

1. We present a fairness evaluation framework using fuzzing called FuzzE. Our framework uses style transfer for text. Intuitively, by using style transfer, we can generate AAVE-like text using only SAE data. Our framework can generate a large number of test cases to evaluate how offensive language classifiers will perform on different linguistic dialects. Moreover, we evaluate the use of multiple style transfer methods to estimate fairness as part of our framework. Finally, we provide a simple, yet effective, approach to ensembling multiple style transfer methods to estimate fairness.
2. We conduct a detailed analysis of the framework using automatic style transfer evaluation metrics. Moreover, we measure the increase of well-known phonetic and syntactic AAVE constructions produced by different style transfer techniques after being applied to SAE text. We also perform a human evaluation study to measure semantic change (e.g., offensive to not-offensive) encountered by transforming the style of text.

2 Related Work

In this section, we describe three major areas of related work relevant to this paper: style transfer, fairness, and offensive language classification.

Style Transfer. Style transfer originates from computer vision, where an image is transformed into a specific artistic style, e.g., an image taken by a cell phone can be made to look like a Van Gogh painting (Gatys, Ecker, and Bethge 2016; Johnson, Alahi, and Fei-Fei 2016). Recently, this idea has been applied to text. For example, many datasets focus on transforming text from a positive sentiment to negative sentiment without changing the underlying topic, or transforming text written from a male's perspective to the perspective of a female. Lample et al. (2019) also transformed text between different age groups, e.g., text written by someone in their 70's is transformed to look as if it was written by a teenager.

In this work, we incorporate state-of-the-art style transfer methods (Li et al. 2018; Prabhumoye et al. 2018) into our framework for the purpose of ranking systems with respect to fairness. We note that as new style transfer methods are developed, they can be applied as a drop-in replacement to the methods discussed in this paper.

Fairness. Fairness is an important topic among natural language processing researchers. Bias has been found in word embeddings (Bolukbasi et al. 2016; Zhao et al. 2018; 2019), text classification models (Dixon et al. 2018; Park, Shin, and Fung 2018; Badjatiya, Gupta, and Varma 2019), and in machine translation systems (Font and Costa-jussà 2019; Escudé Font 2019). In general, each paper focuses on either testing whether bias exists in various models, or removing bias from classification models for specific applications. However, to measure bias and test bias-removal methods, it is necessary to either annotate or infer the demographic information for each user.

Our work is most similar to Shen et al. (2018), where the authors matched words between two genders, races, and political orientations, then analyze how sentiment predictions change by swapping specific words. In their work, the meaning between two words must be the same. However, there are many words that may appear in SAE tweets, but not AAVE. For example, common proper nouns in SAE tweets may not be discussed in AAVE text. We argue that relying on one-to-one translations between word pairs limits our ability to test the robustness of our models. For example, as previous stated, simple lexical fairness estimation methods do not capture subtle differences in the use of prepositions. Offensive language classification methods should be robust to slight changes in preposition usage as long as offensive tweets stay offensive and vise-versa. Moreover, in this work we focus on fairness ranking whereas Shen et al. (2018) studied the impact of prediction changes across small groups of words.

Hate Speech and Offensive Language. Offensive content is a serious concern for social media companies, government agencies, and online communities. The leading approach to handle offensive language online is to flag such content. Many datasets have been collected and annotated for hate speech and offensive language detection on Twitter (Zampieri et al. 2019; Davidson et al. 2017). Likewise, offensive language and hate speech lexicons have been curated to facilitate offensive language detection (Davidson et al. 2017; Wiegand et al. 2018).

Given the recent interest in classifying offensive language, many methods have been proposed to detect it. Razavi et al. (2010) combined naive Bayes with a multi-level classification strategy to detect offensive language. Gambäck and Sikdar (2017) applied convolution neural networks, showing significant improvements over logistic regression with ngram-based features.

Recently, dos Santos, Melnyk, and Padhi (2018) use style transfer to remove offensive language from the text. This is contrary to work that flags offensive content. While similar to our paper, our work differs in the final application. Specifically, we want to generate AAVE-like language from SAE text. The generated tweets should contain offensive words if they were in the original tweet (i.e., the offensiveness should remain the same).

3 Datasets

In this section, we provide context on each dataset that we investigate and describe how they are used for training and

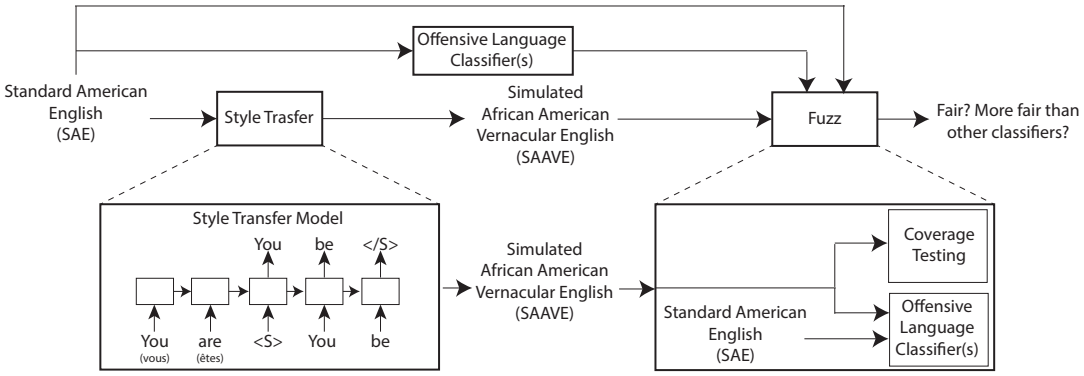


Figure 1: Workflow for the FuzzE framework.

	Style	OLID	HSOL
Training	SAE	10209	9591
Test	SAE	2552	2398
	AAVE	479	2520

Table 1: Number of examples in each split of the two offensive language datasets: OLID and HSOL.

evaluating our offensive language detection models with regard to the FuzzE framework.

AAVE Dataset (StyleData). Blodgett, Green, and O'Connor (2016) originally collected and released more than 59.2 million tweets by 2.8 million users. Each tweet is accompanied with inferred linguistic style information. Following the work by Elazar and Goldberg (2018), we limit our study to all AAVE and SAE tweets with a confidence of at least 80%. This procedure results in 1.6 million AAVE tweets. We also randomly sample 5 million SAE tweets. The datasets reflect "extreme" differences between SAE and AAVE. We hypothesize that this allows us to test unfair "edge cases" of the offensive language classification models. Moreover, we would expect the offensive language classifier's predictions to be similar for SAE and AAVE tweets that do not differ substantially with regard to style and content; however, this needs to be tested.

StyleData is used to train a Convolutional Neural Network (CNN) Kim (2014) to classify tweets as being written with a SAE or AAVE-like style. Furthermore, this dataset is also used to train the style transfer methods that are used as part of our FuzzE framework.

Offensive Language Datasets. We investigate style transfer and fairness evaluation using two datasets: The Offensive Language Identification Dataset (OLID) (Zampieri et al. 2019) and the Hate Speech and Offensive Language (HSOL) Dataset (Davidson et al. 2017). OLID contains 13,240 tweets labeled using a hierarchical annotation scheme where the top level (task A) differentiates offensive and not-offensive tweets. The bottom level (task C) categorizes insults/threats as targeting an individual, group, or other. For the purposes of this paper, we only utilize the first level, task A, of the hierarchy. The first level contains two classes: "Offensive" and "Not Offensive". HSOL contains 14,509 tweets, each labeled with one of three categories: "Hate Speech", "Offensive Language but not Hate Speech", and "Not Offensive". For the purpose of this paper, and to standardize the outputs across both datasets, we group "Hate Speech" and "Offensive Language but not Hate Speech" into a single "Offensive Language" class.

All AAVE inferred tweets—based on the CNN trained on StyleData—are removed from the OLID and HSOL datasets. The AAVE tweets are used for testing. The SAE tweets in both datasets are split into a training (80%) and test set (20%). We use the data to train/test the offensive language classifiers and rank them with respect to fairness.

4 Method

The models used in this paper fall into two groups: style transfer and offensive language classification models. The overall workflow of FuzzE is summarized in Figure 1. Intuitively, we propose a tool that takes an offensive language dataset that only contains SAE text, then transforms the SAE text into simulated AAVE (SAAVE) text with the help of style transfer. Both the SAE and SAAVE text are passed to an offensive language classifier to compare the predictions and assess fairness. We briefly describe the style transfer and offensive language models that are part of FuzzE in the following subsections.

4.1 Offensive Speech Model

For the offensive language classifier, we train a Logistic Regression (LR) model. Specifically, we train an L2 regularized LR model using tfidf-weighted unigrams and bigrams. Using cross-validation, the regularization parameter is optimized for each dataset independently. We found the best regularization parameters for OLID and HSOL to be 0.1 and 1.0, respectively.

4.2 Style Transfer Models

We experiment with four style transfer methods: Back-translation, Retrieval, Template, and an Ensemble. Each method is trained to transform SAE text to be AAVE-like using the StyleData dataset. This section briefly describes

each method we use in our experiments. However, it is important to note that while we present a few methods, as new style transfer techniques are developed, they can be plugged into the FuzzE framework as-is.

Formally, given two datasets $X = \{x_1, \ldots, x_n\}$ and $U = \{u_1, \ldots, u_v\}$ in styles s_1 and s_2, respectively, we learn a model that transforms X into $\hat{U} = \{\hat{u}_1, \ldots, \hat{u}_n\}$ in style s_2. The style-modified data $\hat{U}$ should preserve the semantic meaning of the sentences in X (i.e., offensive text should stay offensive after processing it).

Back-Translation (Prabhumoye et al. 2018). The intuition behind back-translation for style transfer is to develop a representation of the text that (1.) retains the original meaning of the text and (2.) removes, or reduces, the author's stylistic characteristics from the text. Thus we transform the style from SAE to AAVE using a two step approach. First, following the back-translation framework, we translate each tweet from English into French. We found the translation model used in Prabhumoye et al. (2018) to perform poorly on AAVE text. Therefore, we used Google Translate to transform all 1.6 million AAVE tweets in the StyleData dataset to French. French was chosen to align with the original back-translation model (Prabhumoye et al. 2018).

Second, given the translated tweets, we train a sequence-to-sequence model that learns to translate French into a English text with AAVE-like characteristics. Formally, we learn a model $p(u_i|\mathbf{z}_i)$ to map between the two styles, s_1 and s_2, where $\mathbf{z}_i$ represents the vector representation of i-th french-translated example in style s_1. The representation is defined as

$$\mathbf{z}_i = \text{Encoder}(x_i^f; \theta_e) \tag{1}$$

where x_i^f is the Google Translation of the i-th tweet, θ_e is the parameters of the bi-LSTM model, Encoder() represents a bi-direction LSTM model that takes the French Google Translated text and generates a vector $\mathbf{z}_i$. It is important to note differences between training and inference. During training, french-translated tweets in s_2 (AAVE) are used as input to the Encoder. However, at test time, french-translated tweets in s_1 (SAE) are used.

Next, given $\mathbf{z}_i$, the vector representation of the French translation from Equation 1, we train a bi-directional LSTM decoder $\text{Decoder}(\mathbf{z}; \theta_d)$, where θ_d is the parameters of the bi-LSTM model. Furthermore, following Prabhumoye et al. (2018), we use global attention at each step t of the generation processes

$$\alpha_t = \frac{\exp(\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_t))}{\sum_{i \in T} \exp(\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_i))}$$

where $\mathbf{h}_t \in \mathbb{R}^q$ is the bi-LSTM hidden state for the current time-step t of the Decoder, q is the dimension of the hidden states, and $\bar{\mathbf{h}}_t \in \mathbb{R}^q$ represents the Bi-LSTM Encoders hidden state of the source text (i.e., French text). Likewise,

$$\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_t) = \mathbf{h}_t^T \bar{\mathbf{h}}_t$$

represents the similarity between $\mathbf{h}_t$ and $\bar{\mathbf{h}}_t$. For the model specification of the generator and encoder, we use a two-layer Bi-LSTM with a word embedding size of 300 and hidden dimension size of 500. The generator will create a max sequence of 50 tokens.

Retrieval-based Style Transfer (Li et al. 2018). *Retrieval* is a TFIDF-based search method that returns the most similar AAVE sentence using a SAE tweet as the query. Specifically, given a sentence x_i in style s_1, we return the most similar sentence u_j in style s_2. Following Li et al. (2018), we only index *content words*—words that are not indicative of each style. For example, *attribute words* such as "sholl", "iont", and "sumn" (i.e., words common in AAVE (Blodgett, Green, and O'Connor 2016)) are not indexed. Stopwords are also removed from each query sentence. The retrieved sentence is used as the new stylized version verbatim/as-is.

Template-Based Style Transfer (Li et al. 2018). *Template* is an extension of Retreival. Using only content words from the input sentence x_i of style s_1, we find the most similar sentence u_j in the target style s_2. Next, the u_j sentence's attribute words are used to replace the attribute words in the x_i sentence. If the number of attribute words in the retrieved sentence is smaller than the number of attribute words in the query sentence, we use the empty string for subsequent replacements. Refer to Li et al. (2018) for more details.

Ensemble (ENS). Besides the individual models described above, we also evaluate an ensemble method that combines the result of Back-translation, Retrieval, and Template. Specifically, the ensemble is an average of the the fairness metrics we use in Section 5.3: False Positive Equality Difference (FPED) and False Negative Equality Difference (FNED) (Dixon et al. 2018). Intuitively, we are performing model averaging. However, instead of averaging probability outputs, we average the FPED/FNED estimates using synthetic data. FPED and FNED are defined as

$$\text{FPED} = \sum_{t \in T} |FPR - FPR_t| \text{ and} \tag{2}$$

$$\text{FNED} = \sum_{t \in T} |FNR - FNR_t|, \tag{3}$$

respectively, where $T = \{\text{SAAVE}, \text{SAE}\}$ using the synthetic AAVE text generated by a specific style transfer method.[1] FPR and FNR represent the overall false positive and false negative rates, respectively. FPR_t and FNR_t represent the group-specific (i.e., SAAVE or SAE) false positive and false negative rates.

$FPED$ and $FNED$ are calculated independently for each style transfer method by applying the offensive language classifier to each of the style transfer method's generated SAAVE text. The assumption is that the offensive annotations of the original SAE text are still relevant after transforming it into SAAVE (i.e., offensive tweets stay offensive). The ensemble score is calculated as

$$FXED_{ens} = \frac{1}{|M|} \sum_{k \in M} FXED_k$$

where $FXED_k$ represents $FPED$ or $FNED$ for the k-th style transfer method and $M = \{\text{Back-Translation}, \text{Retrieval}, \text{Template}\}$.

[1] For evaluation, we also calculate FPED/FNED scores using the ground-truth AAVE text where $T = \{\text{AAVE}, \text{SAE}\}$.

Style Post-Processing. In many cases, after style transfer, the generated text (SAAVE) loses the offensiveness from the original tweet. Thus, we use a lexicon of offensive words and search for them in the SAE tweet before transforming it.[2] For example, using the lexicon, and given the tweet "You are a bitch," we know that the word "bitch" is offensive. The tweet may be transformed into "Y'all be a female" using style transfer. If the offensive tweet becomes "not-offensive", then it is impossible to estimate fairness without human annotation. To ensure that the tweet is still offensive after processing it, we append the offensive words found in the original SAE tweet to the generated synthetic AAVE text. Thus, "Y'all be a female" becomes "Y'all be a female bitch." Post-processing is applied to all of the style transfer methods.

5 Results

The evaluation strategy focuses on answering three questions: Can state-of-the-art style transfer methods transform SAE tweets into AAVE-like text? If we use style transformed text in place of real AAVE data for evaluation, can we correctly rank the fairest classifiers? Does a better style transform method guarantee better fairness rankings of different models? To answer these questions, we ground our evaluation strategy in the fuzzying evaluation methodology.

Specifically, we use three metrics to evaluate the effectiveness of our framework:

- **Coverage** is generally used to measure how well test cases cover all aspects of a program. For natural language, we quantify how many well-known AAVE characteristics are generated. Moreover, we measure how many AAVE-like tweets are produced. We quantify these coverage quantities in two ways. First, we measure the increase of well-known AAVE phonological variants and syntactic constructions from Blodgett, Green, and O'Connor (2016). Second, we use the CNN trained on StyleData to classify whether a given style transferred string is AAVE-like or not.Via the use of the classifier, we are not constrained to measuring manually curated AAVE linguistic characteristics.
- **Pass Rate**, in Liu et al. (2019), measures how many generated C programs are valid. For natural language, it is a measurement to see how well meaning is preserved after style transfer. For our experiment, we are not interested in the exact semantics of the original tweet being preserved. For example, if a tweet was originally about Backstreet Boys, but after style transfer it mentions Britney Spears instead, this does necessarily hinder our framework. What does matter is that offensive tweets stay offensive and non-offensive tweets stay not-offensive. Unfortunately, this is not possible to measure automatically. Therefore, we use human annotators to measure whether tweets change between offensive and non-offensive.
- **Fairness** estimation is the ultimate goal of FuzzE. For this measure, we analyze how well we can estimate fairness using synthetic data compared to real AAVE text. We discuss the evaluation methodology for this metric in Section 5.3.

[2]http://www.cs.cmu.edu/~biglou/resources/bad-words.txt

	OLID			HSOL		
	O-be-V	**gon-V**	**done-V**	**O-be-V**	**gon-V**	**done-V**
Template	3.50	2.98	2.00	**1.35**	1.95	1.87
Retrieval	**3.88**	5.20	3.56	1.20	**2.76**	**3.47**
Back-Trans.	3.00	**10.61**	**9.00**	0.78	0.62	0.93

Table 2: OLID/HSOL. Ratio of syntactic constructions compared to the original SAE data after style transfer.

	OLID		HSOL	
	% AAVE	**Perp.**	**% AAVE**	**Perp.**
Original	3.75	346.30	21.02	416.32
Template	**68.15**	863.82	**70.65**	890.39
Retrieval	66.36	**567.62**	69.43	551.00
Back-Translation	61.35	642.80	63.37	**510.78**

Table 3: Automatic OLID/HSOL style transfer evaluation. "% AAVE" measures the number of tweets classified as being AAVE-like and "Perp." represents perplexity.

5.1 Coverage

In this section, we explore the style transfer method's "coverage" of AAVE characteristics. Three coverage-based metrics are analyzed. First, we measure the increase of well-known phonetic and syntactic AAVE stylistic characteristics. Next, we apply a classifier that distinguishes between AAVE and SAE text. The classifier is used to automatically measure the increase in AAVE-like characteristics without relying on well-known constructions. Finally, we define a fluency metric, to ensure our models generate realistic text. The results discussed in this subsection (i.e., the numbers in Tables 2 and 3) are calculated using the entire HSOL and OLID datasets (training+test).

Phonetic and Syntactic Alignment. Blodgett, Green, and O'Connor (2016) shown that AAVE language on social media exhibits unique characteristics compared to SAE text. For example, AAVE language contains many phonological variants (e.g., sumn, sholl, and iont). We find that the expression of these variants increases after applying the style transfer methods. On the OLID dataset, compared to the original SAE tweets, "sumn" occurs 9 times more often with Retrieval, 9 times more often with the Template method, and 3 times more often with the Back-Translation method. Similarly, in the HSOL dataset, "sumn" appears 20 times more often after applying Retrieval and Template. Back-Translation does not increase the number of occurrences of "sumn" in the HSOL dataset.

We also analyze three well-known AAVE syntactic constructions (Blodgett, Green, and O'Connor 2016): habitual be, future gone, and completive done. We use a Twitter-specific part-of-speech tagger, Twokenizer (Owoputi et al. 2013) to annotate each tweet. In Table 2, for the OLID dataset, we compare the ratio of each construction in the

	OLID						HSOL					
	Pearson *r*			Spearman rho			Pearson *r*			Spearman rho		
	FPED	**FNED**	**AVG**	**FPED**	**FNED**	**AVG**	**FPED**	**FNED**	**AVG**	**FPED**	**FNED**	**AVG**
Retrieval	.169	.379	.274	.195	.359	.277	.300	.376	.338	.282	.367	.325
Template	**.400**	.526	**.463**	**.375**	.501	**.438**	.221	.415	.333	.210	.427	.318
Back-Translation	.221	.432	.326	.207	.412	.309	.336	.445	.376	.318	.401	.360
ENS	.293	**.555**	.424	.271	**.528**	.400	**.352**	**.509**	**.430**	**.334**	**.493**	**.413**

Table 4: Averaged correlation and ranking results comparing the FPED/FNED scores between SAE and AAVE text with estimated scores using SAE and synthetic (style transferred) AAVE tweets.

	OLID	HSOL
Template	.81	.82
Retrieval	.70	.80
Back-Translation	**.84**	**.91**

Table 5: Pass Rate. Human evaluation of semantic change after style transfer, measuring the % agreement between the human annotators and the original classes.

generated AAVE tweets compared to the original SAE data for the three different style transfer methods. For example, if the O-be/b-V (habitual be) construction appears k times in the original data and h times after processing each tweet using the Back-Translation method, the ratio is defined as $\frac{h}{k}$. All methods produce around 3 times more tweets with the "habitual be" (O-be/b-V) construction. For the "future gone" (gone/gne/gon-V) and "completive done" (done/dne-V) constructions, the Back-Translation method outperforms the other approaches, producing nine times more occurrences in the generated text. The increases on HSOL are not as extreme as seen in the OLID data. Overall, we find that the style transfer methods are able to make the data more AAVE-like based on the in increase of the phonological and syntactic characteristics.

Automatic Style Transfer Validation. Translation metrics such as BLEU (Papineni et al. 2002) have commonly been used for style transfer. Unfortunately, we qualitatively found that a better BLEU score does not translate to better style transfer. In some cases, all of the words may change, while the semantic content stays the same. To evaluate the style transfer methods, we train a binary Convolutional Neural Network (CNN) classifier (Kim 2014) using the StyleData dataset that learns to predict whether a tweet is "AAVE" or "SAE". Intuitively, instead of analyzing every possible syntactic variation in AAVE language, we let a classifier implicitly learn the differences between AAVE and SAE. The CNN classifier is trained with 100 filters that span 5 words. If the style transfer methods generate AAVE-like tweets, then the CNN should classifier them as such. Furthermore, we evaluate the "fluency" of each style transfer method by analyzing the perplexity based on a pretrained KenLM language model (Heafield 2011) trained on a large external Twitter dataset.

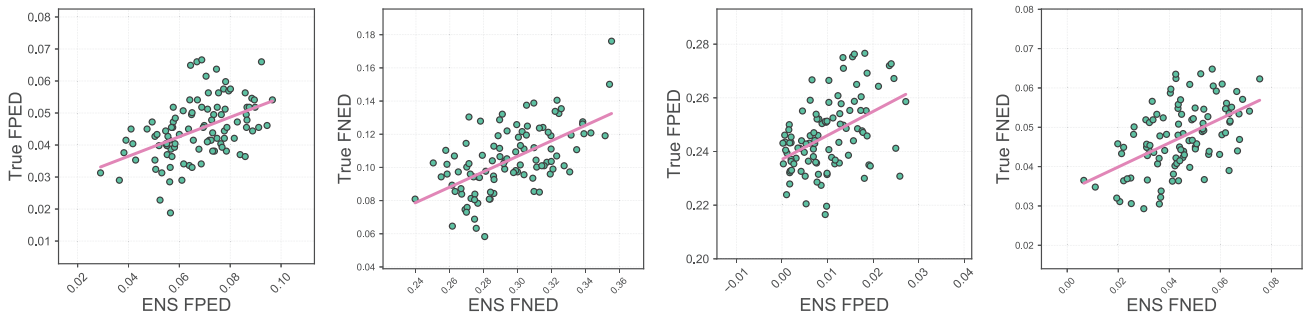
In Table 3, we present the results on the OLID and HSOL datasets. For all three style transfer methods the number of inferred AAVE tweets increases from 3.75% to over 61%—creating 17 times more AAVE tweets than were originally available in the dataset. In terms of fluency, we find that Back-Translation and Retrieval achieve the best performance. This result is expected given Retrieval returns real tweets. Interestingly, the original data has substantially lower perplexity than Retrieval. We believe this is because most of the OLID tweets are in SAE, which is the most prominent language variation on Twitter (Blodgett, Green, and O'Connor 2016). We find similar improvements on the HSOL dataset, with all methods producing more than 60% AAVE-like tweets based on the CNN predictions. Yet, only 21% of the original tweets are classified as AAVE.

5.2 Pass Rate

To measure the pass rate, we perform a human study where a single human annotator sampled 100 tweets from each dataset-model combination, for a total of 600 annotations. We sample the same tweets across each model for a given dataset. Moreover, from the 100 tweets we sample, 50 tweets were originally "Offensive" and the other 50 are were originally "Not-Offensive". The human annotator reviewed the tweets after being processed by the style transfer methods. The annotator then relabels each tweet as either "Offensive' or "Not-Offensive" without looking at the original annotations. Intuitively, we want to make sure that the offensiveness of a tweet does not change after applying style transfer. If the offensiveness changes, then the pass rate drops. In Table 5, we show the results of the human evaluation. We find that all methods generally preserve the "offensiveness" of the original tweets. Back-translation performed the best by having an agreement of 0.84 on OLID and 0.91 on HSOL. Here, agreement measures the proportion of relabeled tweets that match the class of the original text before style transfer.

5.3 Fairness Estimation

Experimental Setup. It is infeasible to develop hundreds of offensive language detection methods to test on each dataset. Instead, using the training split, we create 100 random samples with replacement (i.e., bootstrap sampling) such that each sample contains 60% of the training dataset. We then train the LR offensive language classifier from Section 4.1 on the bootstrap sampled training splits. Using the SAE test data and the real AAVE tweets, we record the



(a) OLID FPED (b) OLID FNED (c) HSOL FPED (d) HSOL FNED

Figure 2: Correlation plots comparing 100 LR models optimized on different subsets of the training data. "True FPED/FNED" represents the real fairness scores, while "ENS FPED/FNED" are the averaged FPED/FNED scores using synthetic AAVE text (SAAVE) from each style transfer method.

FPED and FNED (See Equation 2 and 3) scores for each model. Furthermore, using the real SAE test data and the SAAVE data—the SAE test data transformed into synthetic AAVE data using style transfer—we also record FPED and FNED scores. Next, we create two ranked lists for each metric, FPED and FNED. One list of ranked scores using the real AAVE data, and the other with SAAVE. We compare the two rankings using two correlation metrics: Pearson r and the Spearman rho correlation. Pearson r is a measure of the linear correlation between two variables. Spearman rho measures the strength and direction of the monotonic relationship between two lists.

Results. We repeat the evaluation process 100 times and report the average of each correlation metric in Table 4. Overall, based on the Spearman rho metric, the fairness metric results are positively correlated with real AAVE data. We also visualize the correlations in Figure 2 by analyzing 100 models trained on different subsets of the training data. For both datasets, the ranked FPED/FNED scores are *moderately correlated* with the ground-truth results.[3] Moreover, on average, Template outperforms both Retrieval and Back-Translation on the OLID dataset. Template also performs comparably to Back-Translation on the HSOL dataset. Overall, we find that averaging the fairness estimates across each method results in the most robust estimate of fairness, at least based on the results in Table 4.

5.4 Discussion

In this section, we summarize the results by answering a few questions. First, **can state-of-the-art style transfer methods transform SAE tweets into AAVE-like text?** In Section 5.1, we show that all three style transfer methods produce a substantial increase in well-known phonological (e.g., sumn) and syntactic characteristics (e.g., O-be/b-V). Moreover, in Table 3, we show an increase in AAVE-like text. Yet, given the best method only classified 70% of the SAAVE text as being AAVE-like, there is room for improvement

[3]While not directly related, translation metrics such as BLEU generally have correlations in the range of 0.2 to 0.4 (Reiter 2018).

	Spearman rho w/ post-processing		Spearman rho w/o post-processing	
	FPED	**FNED**	**FPED**	**FNED**
Retrieval	.282	.367	-.312	.269
Template	.210	.427	-.430	.332
Back-Translation	.318	.401	-.531	.214
ENS	.334	.493	-.578	.323

Table 6: Results on HSOL with (w/) and without (w/o) using a lexicon to constrain each method's outputs.

If we use style transformed text in place of real AAVE data for evaluation, can we correctly rank the fairest classifiers? In Table 4, we show that the rank of the FPED/FNED scores using SAE and SAAVE text has *moderate correlation* with the use of SAE and real AAVE text. With only moderate correlation, there is still room for improvement.

Does a better style transform method guarantee better fairness rankings of different models? Not necessarily. In Table 4, we find that ENS perform best overall, followed by Template, then Back-Translation. Retrieval performs worse than the other techniques. Overall, we found that the number of syntactic constructions (Table 2) generated by each of the three major techniques was a toss-up, i.e., different models performed better depending on the construction. However, based on the % AAVE results in Table 3, the Template method outperformed both Retrieval and Back-Translation. With regard to pass rate in Table 5, Back-Translation performed best and Retrieval was the worst. Intuitively, to estimate fairness, we find that it is important for successful style transfer methods to generate a large number of AAVE-like text based on a classifier as well as keeping the original offensiveness of the original tweet. While Retrieval performed well in Table 3 (i.e., the text as AAVE-like), it performs poorly with regard to pass rate which suggests that a balance of AAVE-likeness and pass rate is important. We hypothesize that better controlled generation that can vary the lexical overlap with the original tweet can help improve

Original to ***AAVE***	
SOURCE	No you're a nigger .
Template	No nigger
Retrieval	according to nigger jim #CENSOREDHASHTAG
Back-Translation	No y'all be a nigger
SOURCE	Bobby Flay in this bitch
Template	Ugly bitches favorite line in this I be nawh bitch
Retrieval	Ugly bitches favorite line : I'm far from ugly; I be like nawh bitch , you closer than you think .
Back-Translation	Lil niggas in this bitch
SOURCE	My fucking cousin cracks me up
Template	My her b-day cousin. fucking
Retrieval	On my way! to south street with my cousin to celebrate her b-day!! fucking
Back-Translation	My damn cousin make me crack. fucking

Table 7: Example system outputs using different **SOURCE** sentences. The outputs include the lexicon-based post-processing trick where we append offensive words that appeared in the source text but were not in the generated text.

the fairness rankings. In the meantime, we suggest the use of an ensemble of style transfer techniques which we find to be the most robust across both datasets.

What is the impact of the lexicon-based post-processing applied to style transferred text? It is important that domain knowledge is incorporated into the style transfer models. In this work, we use an offensive language lexicon. We perform a small ablation study by not adding offensive language words missing in the generated output. The results of the study are presented in Table 6. We find that if we do not add missing offensive words, the overall correlation drops substantially for both FPED and FNED. Furthermore, the FPED scores become negatively correlated with the real rankings. Meaning, based on the fairness measures, that best models are ranked in reverse order. Hate speech and offensive language are generally rare in everyday tweets, at least compared to other topics (e.g., sports). Therefore, style transfer methods are prone to remove offensive words from text. This issue is the motivation behind dos Santos, Melnyk, and Padhi (2018), where the authors show that style transfer is a powerful method of fighting abusive speech, because of their property of removing offensive words. Therefore, for the current state-of-the-art style transfer methods, **lexicon-based information should be included for accurate fairness rankings**.

Finally, we look at examples generated by the three style transfer methods explored in this work in Table 7. We show cases where style transfer worked (e.g., Back-Translation translates "Bobby Flay" to "Lil niggas" and "you're" to "y'all be") as well as failures (e.g., Template translates "My fucking cousin" to "My her b-day cousin"). Overall, from a qualitative analysis, Back-Translation does a better job of capturing the original semantic meaning of the tweet. Most of the time, the Template and Retrieval methods only capture the same semantic meaning when the retrieved text discusses the same topic. We also have many cases where the offensive word from the original text is removed by the style transfer technique. Therefore, it needs to be added using the post-processing trick. For example, Back-Translation replaces "fucking" with "damn".

6 Conclusion

Analyzing model behavior across different demographics is important if machine learning is to be used in production systems. This concept is also argued by Mitchell et al. (2019) when creating "model cards". However, if certain demographics are not represented in a dataset, how can we measure fairness? In this paper, we show that style transfer can be used to generate synthetic AAVE text when it is not available in a specific labeled dataset. Overall, while successful to some extent, our main message remains cautionary: **if the application can adversely impact minorities, it is vital to manually annotate real-world minority data to measure fairness**. The goal of this work is to provide model builders the ability to test a large number of models on a new dataset before investing in human annotation for fine-tuning. Because of resources constraints, many developers may never test the fairness of their models.

There are two major avenues for future work. First, more sophisticated methods should be explored to ensure the generated sentences contain specific offensive words. For example, in this work, any missing offensive word that was available in the original tweet, but removed by a style transfer method, is simply added to the end of the generated text. We could incorporate a finite state acceptor which has been used to constrain neural models for poetry generation (Ghazvininejad et al. 2016). Second, we are interested in analyzing fairness in biomedical areas such as public health surveillance where biased systems can adversely effect policy decisions and people's well-being.

Acknowledgments

We would like to thank the reviewers for their insightful comments and help improving this paper.

References

Badjatiya, P.; Gupta, M.; and Varma, V. 2019. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *The World Wide Web Conference*, 49–59.

Bird, D. L., and Munoz, C. U. 1983. Automatic generation of random self-checking test cases. *IBM systems journal* 22(3):229–245.

Blodgett, S. L.; Green, L.; and O'Connor, B. 2016. Demographic dialectal variation in social media: A case study of african-american english. In *Proc. of EMNLP*, 1119–1130.

Bolukbasi, T.; Chang, K.-W.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *NurIPS*, 4349–4357.

Davidson, T.; Warmsley, D.; Macy, M.; and Weber, I. 2017. Automated hate speech detection and the problem of offensive language. In *Proc. of ICWSM*.

Dixon, L.; Li, J.; Sorensen, J.; Thain, N.; and Vasserman, L. 2018. Measuring and mitigating unintended bias in text classification. In *Conference on AI, Ethics, and Society*.

dos Santos, C. N.; Melnyk, I.; and Padhi, I. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proc. of ACL*, 189–194.

Elazar, Y., and Goldberg, Y. 2018. Adversarial removal of demographic attributes from text data. In *Proc. of EMNLP*, 11–21.

Escudé Font, J. 2019. Determining bias in machine translation with deep learning techniques. Master's thesis, Universitat Politècnica de Catalunya.

Font, J. E., and Costa-jussà, M. R. 2019. Equalizing gender biases in neural machine translation with word embeddings techniques. *arXiv preprint arXiv:1901.03116*.

Gambäck, B., and Sikdar, U. K. 2017. Using convolutional neural networks to classify hate-speech. In *Proc. of Workshop on Abusive Language Online*, 85–90.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proc. of CVPR*, 2414–2423.

Ghazvininejad, M.; Shi, X.; Choi, Y.; and Knight, K. 2016. Generating topical poetry. In *Proc. of EMNLP*, 1183–1191.

Hardt, M.; Price, E.; Srebro, N.; et al. 2016. Equality of opportunity in supervised learning. In *Proc. of NeurIPS*, 3315–3323.

Heafield, K. 2011. Kenlm: Faster and smaller language model queries. In *Proc. of workshop on statistical machine translation*, 187–197.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proc. of ECCV*, 694–711.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proc. of EMNLP*, 1746–1751. Doha, Qatar: Association for Computational Linguistics.

Lample, G.; Subramanian, S.; Smith, E.; Denoyer, L.; Ranzato, M.; and Boureau, Y.-L. 2019. Multiple-attribute text rewriting. In *Proc. of ICLR*.

Li, J.; Jia, R.; He, H.; and Liang, P. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proc. of NAACL*, 1865–1874.

Liu, X.; Li, X.; Prajapati, R.; and Wu, D. 2019. Deepfuzz: Automatic generation of syntax valid c programs for fuzz testing. In *Proceedings of the... AAAI Conference on Artificial Intelligence*.

Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I. D.; and Gebru, T. 2019. Model cards for model reporting. In *Proc. of FATML*, 220–229.

Owoputi, O.; O'Connor, B.; Dyer, C.; Gimpel, K.; Schneider, N.; and Smith, N. A. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proc. of NAACL*, 380–390.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, 311–318.

Park, J. H.; Shin, J.; and Fung, P. 2018. Reducing gender bias in abusive language detection. In *Proc. of EMNLP*, 2799–2804.

Prabhumoye, S.; Tsvetkov, Y.; Salakhutdinov, R.; and Black, A. W. 2018. Style transfer through back-translation. In *Proc. of ACL*, 866–876.

Razavi, A. H.; Inkpen, D.; Uritsky, S.; and Matwin, S. 2010. Offensive language detection using multi-level classification. In *Proc. of Canadian AI*, 16–27.

Reiter, E. 2018. A structured review of the validity of bleu. *Computational Linguistics* 44(3):393–401.

Sap, M.; Card, D.; Gabriel, S.; Choi, Y.; and Smith, N. A. 2019. The risk of racial bias in hate speech detection. In *Proc. of ACL*, 1668–1678.

Shen, J. H.; Fratamico, L.; Rahwan, I.; and Rush, A. M. 2018. Darling or babygirl? investigating stylistic bias in sentiment analysis. In *Proc. of FATML*.

Wiegand, M.; Ruppenhofer, J.; Schmidt, A.; and Greenberg, C. 2018. Inducing a lexicon of abusive words–a feature-based approach. In *Proc. of NAACL*, 1046–1056.

Zalewski, M. 2015. American fuzzy lop (afl) fuzzer (2015).

Zampieri, M.; Malmasi, S.; Nakov, P.; Rosenthal, S.; Farra, N.; and Kumar, R. 2019. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

Zhao, J.; Zhou, Y.; Li, Z.; Wang, W.; and Chang, K.-W. 2018. Learning gender-neutral word embeddings. In *Proc. of EMNLP*, 4847–4853.

Zhao, J.; Wang, T.; Yatskar, M.; Cotterell, R.; Ordonez, V.; and Chang, K.-W. 2019. Gender bias in contextualized word embeddings. *arXiv preprint arXiv:1904.03310*.

Zliobaite, I. 2015. A survey on measuring indirect discrimination in machine learning. *arXiv preprint arXiv:1511.00148*.