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

Paper ID: 9225

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). 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 fairness on AAVE text using only text written in Standard American English. We quantitatively validate our methodology and provide general advice on analyzing the bias of text classifiers when data for particular minority dialects are unavailable in both the training and test datasets. 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 different demographics, or worse, data for certain groups may simply be unavailable or limited for certain topics.

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”, “My husband”, and “My wife”) in a tweet without changing its meaning. If the models’ prediction changes by these modifications, then we assume the model is biased.

Typically, fuzzing techniques for software testing use blackbox random testing (Zalewski 2015). Yet, the recent fuzz testing approach for fairness relies on manually created templates. Therefore, current approaches only capture single word lexical biases and lack in “bias recall” because of manual curation. For example, 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—depending on the application. This characteristic is not something that would be obvious to try. Therefore, manually capturing all stylistic characteristics is challenging. Moreover, the characteristics may change over time, further increasing the difficulty of manual curation.

Recently, blackbox sequence-to-sequence neural networks have been used to generate fuzzing test cases to find bugs in C programs (Liu et al. 2019). In this work, we investigate the use of neural network-based 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 styles. 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 fuzzy fairness framework called FuzzE that uses style transfer for text. 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 test how offensive language classifiers will perform on different linguistic dialects.

2. We conduct a detailed analysis of the framework using automatic style transfer evaluation metrics. Moreover, we measure increase of well-known phonetic and syntactic AAVE constructions produced by different style transfer techniques after being applied to SAE text. We also perform human evaluation to measure semantic change (e.g., offensive to not-offensive) encountered by transforming the style of text. Overall, we find moderate correlation of fairness rankings between synthetic and real AAVE data.

## 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 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 written with a positive to negative sentiment, or text written from a male’s perspective to the perspective of a female.\(^1\) 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 modify existing state-of-the-art style transfer methods (Li et al. 2018; Prabhumoye et al. 2018) 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 a hot 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 1-to-1 translations limits our ability to test the robustness of our models. Models should be robust to slight changes in topical content, as long as offensive tweets stay offensive, and vice-versa. Moreover, in this work we focus on fairness ranking whereas Shen et al. (2018) studied the impact of model changes across various groups of words.

### Hate Speech and Offensive Language

Hate speech is a serious concern for social media companies, governments, 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 detecting offensive language, many methods have been proposed. Razavi et al. (2010) combined naive Bayes and 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.

Our work is most similar to dos Santos, Melnyk, and Padhi (2018). 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. Our work differs in the final application. Our goal is to generate AAVE language from SAE text. Moreover, we want generated tweets to contain offensive words if they were in the original tweet. Our ultimate goal is to compare the relative fairness of different classifiers for offensive language detection between different demographics.

## 3 Datasets

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

\(^1\)We note that this is an overly simplified view of gender. We are simply stating prior work in this area.
accompany 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 SAE and AAVE tweets that are similar to not change model predictions much, however, this needs to be tested.

This dataset is used to train a Conventional Neural Network (CNN) to differentiate SAE and AAVE text. Furthermore, this dataset is also used to train the style transfer methods.

**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. For the purposes of this paper, we only utilize the first level, or 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.

We assume that each dataset does not contain any AAVE tweets. Therefore, all AAVE inferred tweets—based on the CNN in Section 5.2—are removed from the datasets. Both datasets are split into a training (80%) and test set (20%) using the remaining SAE tweets. These datasets are simply used to train the offensive language classifiers and evaluate fuzzy fairness estimation.

### 4 Method

The models used in this paper fall into two groups: style transfer and offensive speech 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 each the style transfer and offensive language models in the following subsections.

#### 4.1 Style Transfer Models

In this section, we describe the back-translation-based style transfer method which is based on the work by Prabhumoye et al. (2018). We note that any style transfer method can be used in our framework. We present one solution for approach, but as new methodologies are developed, they can be plugged into FuzzE as-is. Formally, given two datasets $X = \{x_1, \ldots, x_n\}$ and $\mathcal{U} = \{u_1, \ldots, u_v\}$ in styles $s_1$ and $s_2$, respectively, we learn a model that transforms $X$ into $\hat{U} = \{\hat{u}_1, \ldots, \hat{u}_v\}$ in style $s_2$. The modified data $\hat{U}$ should preserve the semantic meaning of the sentences in $X$ (i.e., offensive text should stay offensive).

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 for 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 an English text with AAVE-like characteristics. Formally, we learn a model \( p((u_i|z_i)) \) to map between the two styles, \( s_1 \) and \( s_2 \), where \( z_i \) represents the vector representation of \( i \)-th french-translated example in style \( s_1 \). The representation is defined as:

\[
z_i = \text{Encoder}(x_i^f; \theta_e)
\]

where \( x_i^f \) is the Google Translation of the \( i \)-th tweet, \( \theta_e \) is the parameters of the bi-LSTM model, \( \text{Encoder()} \) represents a bi-direction LSTM model that takes the French Google Translated text and generates a vector \( z_i \).

Next, given \( z_i \), the vector representation of the French translation from Equation 1, we train a bi-directional LSTM decoder \( \text{Decoder}(z; \theta_d) \), where \( \theta_d \) 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:

\[
\alpha_t = \frac{\exp(\text{score}(h_t, \overline{h}_t))}{\sum_{i \in T} \exp(\text{score}(h_i, \overline{h}_i))}
\]

where \( h_t \in R^q \) is the bi-LSTM hidden state for the current time-step \( t \) of the Decoder, \( q \) is the dimension of the hidden states, and \( \overline{h}_t \in R^q \) represents the Bi-LSTM Encoders hidden state of the source text (i.e., French text). Likewise, \( \text{score}(h_t, \overline{h}_t) = h_t^T \overline{h}_t \) represents the similarity between \( h_t \) and \( \overline{h}_t \). For the model specification, for the generator and encoder, we use a two-layer Bi-LSTM with a word embedding size of 300 and hidden dimensions of 500. The generator will create a max sequence of 50 tokens.

**Lexicon Constraint.** We modify the output of the style transfer model using an offensive language lexicon. Specifically, we add offensive words to the target text if the word was in the original tweet and lexicon, but not in the style transformed text. For example, if the offensive tweet “You are a b*tch” is transformed into “Y’all be a dog”, we would modify the new text to include b*tch, e.g., “Y’all be a dog b*tch”.

### 4.2 Offensive Speech Model

In Figure 1, for the FuzzE workflow, there are two major models: the style transfer model and an offensive language classifier. 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 on the training OLID and HSOL corpora, the regularization parameter is optimized for each dataset independently. We found the best regularization parameter four OLID and HSOL to be 0.1 and 1.0, respectively.

## 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 fuzzy testing evaluation methodology by connecting it to style transfer metrics.

Specifically, in this study, 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. Overall, we analyze these coverage quantities in two ways. First, we measure the increase of well-known AAVE phonological variants and syntactic constructions. These AAVE stylistics characteristics were previously studied in Blodgett, Green, and O’Connor (2016). Second, we use the CNN trained on StyleData to classify whether a given string has an AAVE-like style or not. This automated coverage test is commonly used to evaluate style transfer methods. Moreover, via the use of the classifier, we are not constrained to measuring manually curated AAVE linguistic characteristics.

- **Pass Rate**, in Liu et al. (2019) measured how may generated C programs are valid. For natural language, it is a measurement to measure 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 non-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 5.4.

### 5.1 Baselines

We compare Back-translation to three other style transfer methods: Retrieval, Template, and an Ensemble. Each method is trained using StyleData. We briefly describe each method below.

**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. Attribute words and stopwords are also removed from each query sentence. The retrieved sentence is used as the new stylized version verbatim.

**Template-Based Style Transfer** (Li et al. 2018). Template is an extension of Retrieval. 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.

**Ensemble (ENS).** For fairness testing, we also compare an ensemble version that combines the Back-translation, Retrieval, and Template methods. For this technique, we generate FPED and FNED fairness metrics using each style transfer method (see Section 5.4), then we average them across the three methods.

### 5.2 Coverage

In this section, we explore the “coverage” of the style transfer methods for fuzzy testing. Two 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.

**Phonetic and Syntactic Alignment.** Blodgett, Green, and O’Connor (2016) show 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” on 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 generated AAVE tweets compared to the original SAE data for the 5 different style transfer methods. For example, if the O-be/h-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

$$\text{ratio} = \frac{h}{k}.$$

All methods produce around 3 times more tweets with the “habitual be” construction. For the “future gone” and “complective done” constructions, the Back-Translation method outperforms the other approaches, producing nine times more occurrences in the generated text. The increases on HSOL, in Table 3, are not as extreme as seen in the OLID data. Overall, based on the increase in AAVE syntactic constructions, the style transfer methods are able to convert SAE text to AAVE—or at least make the test more AAVE-like.

**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 classify 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 Twitter data.

In Table 4, we present the results on the OLID and HSOL datasets. For all 5 style transfer methods the number of inferred AAVE tweets increases from 3.6% 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

| Method         | O-be/h-V | gone/gne/gon-V | done/dne-V |
|----------------|----------|----------------|------------|
| Template       | 3.50     | 2.98           | 2.00       |
| Retrieval      | 3.88     | 5.20           | 3.56       |
| Back-Translation| 3.00     | 10.61          | 9.00       |

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

| Method         | O-be/h-V | gone/gne/gon-V | done/dne-V |
|----------------|----------|----------------|------------|
| Template       | 1.35     | 1.95           | 1.87       |
| Retrieval      | 1.20     | 2.76           | 3.47       |
| Back-Translation| 0.78     | 0.62           | 0.93       |

Table 3: HSOL. Ratio of syntactic constructions compared to the original SAE data after style transfer.
It is important to note that human annotators were originally "Offensive" and the other 50 were originally "Not-Offensive". It is crucial to ensure 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. Overall, 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.

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

| Method         | OLID % AAVE Tweets | Perp | HSOL % AAVE Tweets | Perp |
|----------------|--------------------|------|--------------------|------|
| Original       | 3.62               | 346.30 | 17.37              | 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 5: Pass Rate. Human evaluation of semantic change after style transfer, measuring the % agreement between the human annotators and the original classes.

lower perplexity than Ret. We believe this is because most of the OLID tweets are in SAE, which is the most prominent language variation in "General Data". 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 17% of the original tweets are classified as AAVE. We also note that only 3.6% of the original OLID tweets are classified as AAVE. We expect that the small percentage of original AAVE tweets is the cause of the differences in Tables 2 and 3. Because the proportion of AAVE tweets is small in the OLID dataset, only 3.6%, the increase in AAVE syntactic constructions is higher than the HSOL dataset (17.3%).

5.3 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". It is important to note that human annotators were originally "Offensive" and the other 50 were originally "Not-Offensive". It is crucial to ensure 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. Overall, 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.

5.4 Fairness Estimation

Experiment Setup. Two common metrics for analyzing the fairness of text classifiers are the False Positive Equality Difference (FPED) and False Negative Equality Difference (FNED) (Dixon et al. 2018) as defined as

$$FPED = \sum_{t \in T} |FPR_t - FPR_T|$$
$$FNED = \sum_{t \in T} |FNR_t - FNR_T|,$$

respectively, where $T = \{AAVE, SAE\}$ using the ground-truth AAVE text, or $T = \{SAAVE, SAE\}$ using the synthetic AAVE text generated using style transfer. FPR and FNR represent the overall false positive and false negative rates, respectively. $FPR_t$ and $FNR_t$ represent the group-specific false positive and false negative rates.

It is infeasible to develop hundreds of 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 4.2 on the bootstrap sampled training splits. Using the SAE test data and the real AAVE tweets, we record the FPED and FNED 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.
generally have correlations in the range of 0.2 to 0.4 (Reiter 2018).

For both datasets, 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. 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 6.

### 6 Limitations and Discussion

In this paper, we presented a new task that analyzes style transfer for fairness evaluation. We show that style transfer methods can successfully produce AAVE phonological and syntactic construction from SAE data. 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. Moreover, because of resources constraints, many developers may never test the fairness of their models. Fuzzy testing with FuzzE requires minimal investment.

Differences between the real AAVE text and style transformed (e.g., Back-Translation) versions are subject to covariate shift. There is no guarantee that the data distribution between the original AAVE tweets and generated text should be the same. Furthermore, while many sentences using the Back-Translation model are successfully transformed from SAE to AAVE, e.g., “No you’re a ni**er” to “No y’all be

a ni**er”, there are cases in which the original meaning of the message is lost, e.g., “Cameron just cracks me up all the time” is transformed into “bae make me sunlight all the time”. In future work, we plan to examine techniques to constrain the semantic meaning between the translations. However, in this study, semantic shift does not cause a substantial drop in correlation. This result is expressed by the positive correlation of the Retrieval method. Retrieval simply finds the most similar AAVE tweet given a SAE tweet as a query. There is little chance that the new tweet discusses the same topic as the original SAE text. Yet, it does help to constrain the meaning, given Back-Translation and Template outperform Retrieval on the OLID dataset.

### 7 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 proposed the use of style transfer to generate pseudo AAVE text when AAVE tweets are not available in a specific training dataset. The main assumption is that it is feasible to gather data of the minority demographic of interest. The retrieved data does not need to be relevant to the specific topic (e.g., offensive language detection), nor does it need to be labeled. We constrain the offensive language class via the use of prior information (i.e., a lexicon of offensive words).

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. For example, we could incorporate a finite state acceptor which has been used to constrain neural models for poetry generation (Ghazvininejad et al. 2016). Second, the use of style transfer can be applied to other domains. Specifically, we are interested in biomedical areas such as public health surveillance where biased systems can adversely affect policy decisions and people’s well-being.

---

### Table 6: Averaged correlation and ranking results comparing the FPED/FNED scores between SAE and AA VE text with estimated scores using SAE and pseudo (style transferred) AAVE tweets.

|                      | OLID |              |           | HSOL |              |
|----------------------|------|--------------|-----------|------|--------------|
|                      | Pearson $r$ | Spearman rho | Pearson $r$ | Spearman rho |
| FPED                 | FNED | AVG          | FPED      | FNED | AVG          |
| Retrieval            | 0.169 | 0.379 | 0.274 | 0.195 | 0.359 | 0.277 | 0.300 | 0.376 | 0.338 |
| Template             | 0.400 | 0.526 | 0.463 | 0.375 | 0.501 | 0.438 | 0.221 | 0.415 | 0.333 |
| Back-Translation     | 0.221 | 0.432 | 0.326 | 0.207 | 0.412 | 0.309 | 0.336 | 0.445 | 0.376 |
| ENS                  | 0.293 | 0.555 | 0.424 | 0.271 | 0.528 | 0.400 | 0.352 | 0.509 | 0.430 |

---

While not directly related, translation metrics, such as BLEU generally have correlations in the range of 0.2 to 0.4 (Reiter 2018).
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 NurtIPS, 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 Politecnica 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.; Prajapat, 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.

Prabhhumoye, 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.