Mitigating Racial Bias in Social Media Hate Speech Detection

Jiangxue Han

Advisor: Jane Chandlee

Second Readers: Amanda Payne, Alvin Grissom Jr.

Abstract

New ways of using language emerge in social media. While there are many positive aspects, it also leads to anti-social behavior, cyberbullying, online harassment, and hate speech. As a result, hate speech detection models are often used to recognize hate speech on social media and thus enable platforms to regulate the accounts that show such behavior. In this paper, I will establish that bias against users using African American English (AAE) exist in hate speech detection models and provide a literature review on current approaches to reduce such bias. I then propose to perform lexical and syntactic alternations to remove protected attributes of AAE before training and use an adversarial approach for training to generate hate speech predictions while mitigating racial bias.

Table of Contents

1 Introduction

Hate speech on social media is a common phenomenon, and there are existing models to perform hate speech detection. The use of machine learning models to perform such actions is necessary because it is not realistic to examine all online contents manually. However, it has been shown that bias exists for these models (Sap et al., 2019; Davidson et al., 2019). One group that has experienced such bias is speakers of African American English dialect, or abbreviated as AAE.

African American English does use a particular set of vocabulary that is different from non-AAE speakers, which might contribute to bias. Often times, the same expressions in AAE face higher chance of being labeled as hate speech. An example from Sap et al. (2019) is shown below:

(1) a. Wussup, n*gga!

b. What's up, bro!

c. I saw that n*gga's ass yesterday

d. I saw that dude yesterday

Here the first two sentences convey the same meaning, as well as the last two sentences, given context that one is AAE speaker and the other is non-AAE speaker.

In order to find ways for models to reduce bias and improve accuracy, I would like to explore the question of how language is perceived as offensive and hateful for humans so that I can try to reduce the gap between hate speech detection model and how it is actually received by humans. For example, it is argued that offensiveness is not necessarily communicated by the

contents of language but depends on what subjective image the offender is trying to construct, which often depends on tones or dialectal knowledge (Anderson and Lepore, 2013). In addition, traditionally hateful language can be used within a close group to obtain a humorous effect or can even be used to communicate positive sentiments. These are some factors we might want to consider when improving hate speech detection models.

I propose that we perform alternations to text to remove some AAE related features before sending data into training so as to minimize the influence of dialect. The goal of performing such alternations is that we want to prevent the model from associating AAE features with hate speech and therefore make unfair predictions. After removing AAE features, the model will be forced to learn other features to make hate speech predictions. If this is successful, when we use this model to make hate speech predictions on actual text, it will have less bias against people who speak AAE, thus they will be able to speak their dialect freely on the internet. I am proposing to perform such alternations at data level before feeding to model so that we can reduce racial bias. This is in no way asking to remove dialectal information for user's actual posts. On the contrary, this is enabling users to use AAE dialect more on social media. Such job can be done in a lexical way – perform word swapping to map AAE lexical items into non-AAE alternatives with similar meaning. I will also look into syntactic features of AAE and remove them while preserving the same meaning. This process is only for the purpose of reducing the influence of dialect for hate speech detection models in the intermediate steps of training. The "translated" text is not meant to replace the original AAE text. I will use adversarial training to reduce racial bias, where I will have an encoder that encodes corpora text into high dimensional vector, and an adversary that predicts if a vector was original AAE text. These two models will

train against each other, and the goal of the encoder is to generate vectors that "fools" the adversary. Then I will have a classifier that performs hate speech detection.

2 Literature Review

2.1 Existence of racial bias in social media hate speech detection

To address the issue of racial/dialectal bias in hate speech detection, we need to first establish that such bias exists. Sap et al. (2019) mentioned that widely used hate speech detection datasets have bias in toxicity ratings against African American English. This bias is then acquired and propagated through trained model on corpora so that tweets from African Americans are up to two times more likely to be labelled as offensive. The authors used DWMW17 (Davidson et al. 2017), which includes annotations of 25K tweets as hate speech, offensive, or none. In addition, they also used FDCL18 (Founta et al. 2018), which collects 100K tweets annotated with four labels: hateful, abusive, spam or none. The authors analyzed the Pearson Correlation r with proportions of AAE and each toxicity category. Two categories that especially draw attention are "offensive" in DWMW17 and "abusive" from FDCL18, with r value as 0.42 and 0.35 respectively (Sap et al 2019, p. 1670). They further trained a classifier with GloVe vectors (Pennington et al. 2014) that minimizes the cross-entropy of the annotated class conditional on text. The false positive rate of identifying offensive speech for AAE is 46.3% for DWMW17, which is much higher than the 9.0% for White. For FDCL18, false positive rate of identifying abusive speech is 26.0% for AAE, in contrast to 4.5% for White (p.1670). These are evidence that hate speech labeled data is biased against African Americans, and this bias is propagated through models. This will lead to the unfair suppressing of the voice of a minority community that is already disadvantaged in many aspects. People who use AAE

should be able to express themselves and engage in online conversations just like everyone else. Therefore, this calls for immediate action to reduce such bias.

Davison, Bhattacharya, and Weber (2019) published their finding that hate speech detection is biased against African Americans. They trained a classifier to predict the class of unseen tweets for each widely used Twitter datasets using regulated logistic regression with bag-of-words features. Bag-of-words is the modelling approach of characterizing a text as a multiset, ignoring grammatical features but keeping track of the number of times each word appears in the multiset. They used the basic approach of stemming tokens, constructing n-grams, using grid search to find best parameters and a 5-fold cross validation (p.27). Stemming token is the action of converting words into their root form to avoid redundancy. For example, "bought" would be converted to "buy". Grid search allows the authors to test different combinations of hyperparameters and evaluate the test results to find the best parameters. Cross validation is the technique of dividing data into k groups (in this case $k = 5$) and using all other groups except one as training data, and that one last group as validation data, and rotate the group that is used for validation. The model is not intended to improve performance on hate speech detection itself but aims to be a standard baseline model to provide insights into whether bias against African Americans exist. They tested against the null hypothesis that there is no statistical evidence that black-aligned tweets are more likely to be classified as offensive. They used $p = 0.001$ as standard for high statistical significance, and as a result all but "racism" label in Waseem (2016) are statistically significant evidence that bias exists against African Americans (p.29).

2.2 How offensive language is perceived by humans?

In order to understand why hate speech detection models are biased, we need to investigate how language is perceived to be offensive or hateful for humans. The gap between

how humans perceive offensive language and how detection models work might lie potential improvements we can experiment on.

Anderson and Lepore (2013) raised questions on why slurs are offensive to their target members. Is it because of what they semantically express or how they are conventionally implicating? They explain by arguing that slurs are not offensive because of the contents that are communicated, but rather "relevant edicts surrounding their prohibition" (p.3). Therefore, often times the offensiveness comes from the subjective image of what the offenders are constructing, which can rely on tones used or dialectal knowledge.

An interesting observation is that offensiveness is sometimes not transferred through indirect speech. Below are examples from Anderson and Lepore (2013):

(2) A b*tch ran for President of the United States in 2008.

(3) Eric said that a b*tch ran for President of the United States in 2008.

We can observe that (2) is an offensive sentence and conveyed the author's sexist offence. (3) is an indirect speech stating that someone named Eric said this offensive sentence. Even though the main semantic content is the same, Anderson and Lepore believe that (3) does not convey information that its author is making a sexist offense. Whether indirect speech bound offensiveness is a controversial topic. In the case of sentence (3), I believe it's up for listener's interpretation and also depends the tone and emphasis when this sentence is said. If the narrator is barely restating that fact that Eric said so, then they are not agreeing with Eric. However, if they emphasizes the slur "b*tch", it implies that they agree with the use of term and believes that the person referred to should be called with such a slur, then it is still offensive. This is a

complicated situation where simple lexical detection will not be able to accurately capture potential offensiveness.

Anderson and Lepore also observed that slurs could be used in a sentence with positive connotation. Consider the following examples:

(4) He played like a n*gger.

(5) I love wops; they are my favorite people on earth.

For the context, sentence (4) is used to praise a basketball player for performing well. Even though they contain words that are typically considered hateful, they communicate praise and friendly sentiments. They cited Grice that even though some sentences are truth-conditionally equivalent, they might have different semantic values (1961,1989). The example sentences Grice used are below:

(6) John is British but brave.

(7) John is British and brave.

(6) has the presupposition that British are not usually brave, and that is why John being brave as a British is uncommon. As we can tell, this can potentially be offensive to British people. (7) does not have the same meaning. This is another type of subtle feature that is hard for machine learning models to detect.

Another perspective to look into offensive language is through analysis of presupposition and conventional implicature, as discussed by Camp (2018). Camp argued that what makes slurs different from their neutral counterparts is their derogating presuppositions. Then she proceeded to analyze if conditionals can quarantine the degrading presuppositions. She argued that if the antecedent part of the conditional entails the presuppositions that is triggered in the consequent

clause, then the quarantine works and the presuppositions cannot be projected out from the conditional. An example that she used is below:

(8) I consider John a saint. But if he ever screws me over, I'll crush the bastard like a bug!

The offensive message of "crushing the bastard like a bug" only holds if "John screws me over", and the oftensiveness does not escape the conditional. However, this is not entirely true of all conditionals. Consider the following sentence from Camp (2018):

(9) I think Jews are awesome; some of my best friends are Jewish. But if the new hire is a Jew, then they'll regret hiring a k*ke.

Even though the author lays out positive opinions in the beginning, this sentence still clearly adopts the derogating perspective of the racial slur. Camp believes that this is because of the relevant entailment between the antecedent and the consequent. I find the most difference between (8) and (9) lie on the nature of their antecedent – the antecedent of (8) involves "John screwing me over", which is an active action from John. However, the new hire being a Jew in (9) is a racial attribute, and thus should not be used as antecedent where the consequent shows derogading presumptions. This is an example of a subtle semantic difference that is hard to be captured by a model.

Camp also discussed the conventional implicature view, which claims that truth condition is irrelevant to perspective and offensiveness (2018). This suggests that slurs are truth-conditionally equivalent to their neutral counterparts. An intriguing idea Camp proposes is that offensiveness lies in optionality. This means that the existence of the neural counterpart is what makes the use of the slur offensive, when the author could have chosen the one without degrading connotations. There is also discussion on the focus of a sentence. Some words might

be slurs or have negative and offensive connotations in general, but are not the primary point of the sentence. An example given by Camp is (10):

(10) Damn! My f*cking cell phone is on the f*cking fritz again

This sentence has multiple swear words that can be considered offensive in general. However, because the primary point is to express the author's feelings about the phone on fritz instead of trying to attack anyone, this is not offensive.

Babou-Sekkal (2012) also studied how taboo language is perceived by humans in her dissertation. She argued that offensive words in a comedy setting within a group can have humorous effect and intensify the emotional bonds among group members. The "insider knowledge" within the group can change how taboo language is understood. Euphemism is another linguistic phenomenon that can trick detection models. People often use euphemisms to hide unpleasant and offensive intentions, while the same idea is communicated. In addition, the amount of offensiveness is also dependent on the receiver side of the conversation. The same language can be much more hurtful and offensive if the listener is vulnerable and sensitive. A concept Babou-Sekkal brought up is that using a shared language does not mean sharing socio-linguistic rules. Thus, people who speak the same language but come from different backgrounds might misunderstand each other. Similarly, people who speak different dialects of English have different expressions that might confuse each other as well as hate speech detection models. These are all factors that are very difficult for machine learning models to take account of.

2.3 Approaches to mitigate bias

Some work has been done to explore reasons behind racial/dialectal bias in hate speech detection and to reduce such bias (Sap et al., 2019, ElSherief et al., 2018, Xia et al., 2017, Waseem, 2016). These works provide examples of directions to navigate.

ElSherief et al. (2018) focused on the target of offensive speech—whether it's *directed* or *generalized*. *Directed* hate speech targets individuals, whereas *generalized* hate speech targets a population sharing a protected characteristic. The authors used psycholinguistic lexicon software LIWC2015 (Pennebaker et al. 2015) to analyze the psycholinguistic processes and linguistic dimension of directed and generalized hate speech. This tool measures psychological dimensions of languages, such as affection and cognition. SAGE (Eisenstein, Ahmed, and Xing 2011), a supervised model that considers background features and topic, etc., is also employed. The authors found out that directed hate speech is often more informal, angry and involves name-calling, while generalized hate speech is dominated by religious hate and involves use of lethal words, such as murder and kill. This paper lays the groundwork that the characteristics of hate speech is highly related to the target, which leads to further hypothesis that training models with labels for targets of hate speech could improve the accuracy of hate speech detection.

Sap et al. (2019) proposed racial and dialectal priming as a way to reduce racial/dialectal bias in hate speech detection. They ran a controlled experiment to encourage annotators to consider if a particular tweet is offensive to them and/or anyone, and also infer racial background of author. When rating offensiveness to anyone, the mean for control condition (0.55) differs from dialect (0.44) and race (0.44) conditions significantly ($p < 0.001$) (p.1671). Waseem (2016) also proposed a way to improve on the data annotation level. He provided evidence that amateur annotators are more likely to label tweets as hate speech than expert annotators, and models trained on datasets with expert annotations perform better.

In addition to mitigating the bias in annotation level, studies have been done on improving models themselves. Mozafari et al. (2020) used a transfer learning approach using pretrained model Bidirectional Encoder Representations from Transformers (BERT), and proposed a bias alleviation mechanism that looks for explicit bias and studies phrases in training set contributing to such bias. A reweighting mechanism is then proposed to smooth the correlation between such phrases and the classes they belong to. Transfer learning is the process of using existing pre-trained knowledge in one area and apply it to a different but related area. Adjustments and tuning are necessary during this process for the model to be more applicable to the new task. This technique is especially useful in scenarios where we don't have enough training data. BERT is a model trained by Google on general domain corpus, the authors then proceeded to tune it on two common Twitter corpuses - Waseem and Hovy (2016) and Davidson et al. (2017). Davidson et al. (2017) labeled text into three categories – Hate, Offensive, Neither, and Waseem and Hovy (2016) labeled text into Racism, Sexism and Neither.

Mozafari et al. (2020) introduced their bias alleviation mechanism by first identifying high frequency n-grams in each class. This is achieved by calculating local mutual information between each n-gram and class. The following formula is used by the authors to calculate LMI:

$$\mathrm{LMI}(w, c) = p(w, c) \cdot \log\left(\frac{p(c \mid w)}{p(c)}\right) \quad (1)$$

Here *w* represents the n-gram and c represents a class. Some examples of high frequency bigrams for "Racism" class in Waseem dataset found out this way are "muslims are", "prophet muhammed", "pedophile prophet", etc. The same for "Sexism" class includes "sexist but", "but women", "feminazi", etc (p.18). We can tell that some of these phrases don't necessarily imply racism or sexism, but the high correlation is likely going to lead the model to learn it and show

bias. The authors employed Schuster et al.'s algorithm to debias by reweighting the samples (2019). The high frequeny n-grams can be constrained by defining a positive weight α^i for each sample x^i in a way that the importance of tweets containing these phrase but with a different label is increased (p.18). The bias towards a class c is defined by Mozafari et al (2020) as follows:

$$b_j^c = \frac{\sum_{i=1}^{n} I_{[w_j^{(i)}]}\left(1 + \alpha^{(i)}\right) I_{[y^{(i)}=c]}}{\sum_{i=1}^{n} I_{[w_j^{(i)}]}\left(1 + \alpha^{(i)}\right)} \quad (2)$$

Here y^i is the class label and each training n-gram is w_j. This is intuitive because we are looking at the percentage of tweets where the label y is class c. To minimize the bias, the authors finds the weight vector α by solving the optimization problem:

$$\min\left(\sum_{j=1}^{|V|} \max_c\left(b_j^c\right) + \lambda \parallel \vec{\alpha} \parallel_2\right) \quad (3)$$

After optimization is done to minimize bias, the new re-weighted scores for each sample are fed into pre-trained BERT model for fine tuning. The authors found out that after their re-weighting mechanism the probability of black-aligned tweets being classified as racism significantly decreased. *p(black)/p(white)* decreased by 6.8 times for Waseem dateset, indicating validity in the bias alleviating mechanism.

Xia, Field and Tsvetkov (2017) proposed to use the adversarial training to mitigate bias against AAE. They trained a classifier that learns to detect toxic language while demoting the model from learning elements related to AAE. The authors used an encoder to encode the texts into higher-dimensional space, trained a binary classifier that decides if text is hate speech from input, and used an adversary that predicts the protected attribute (AAE in this case) from text. For this task a bidirectional LSTM encoder is chosen, and the classifiers are 2-layered multi-

layer perceptron with *tanh* activation function (p.8). For more detailed explanation of these terms, see section 3.5 and 3.6. Results show that for some datasets their model is able to reduce dialectal information from encoded representation because the accuracy for AAE prediction is declining, especially in the early epochs. However, after certain threshold of epochs the overall accuracy for hate speech detection is also declining, suggesting a tradeoff between hate speech detection accuracy and racial bias removal.

3 Methodology

The literature review that I've done so far focus on either data labelling or using mathematical and/or machine learning approaches to reduce bias. It's not practical within the scope of this thesis project to collect new labeled data. I will build my work on Xia, Field and Tsvetkov (2017). They are protecting the model from learning AAE features through training algorithms, while I propose to remove such features in training data as well. I realize that I can look from a linguistic perspective and try to minimize the effect of dialect on hate speech detection task. It is possible to identify some lexical and syntactic features of AAE and attempt to remove these features in the intermediate steps before sending data to models. I would like to adopt an approach that combines linguistic AAE feature removal and adversarial training that reduces dialectal information in encoded representation of text data. After implementing this project design, we will be able to see if this new approach that involves removing dialectal features is efficient in demoting racial bias against AAE speakers in automatic hate speech detection models.

3.1 General Architecture

I will use the approach of adversarial training to demote using protected features of AAE to perform hate speech detection. I will perform sequential search to remove as much AAE

related lexical and syntactical features as possible before inputing the text into encoder. I will have two models where the first one includes AAE feature removal and the second skips this part so that I can compare the performance later. The architecture will include three parts:

- An encoder *H*, which attempts to eliminate AAE related features and encode text into a higher dimensional vector.
- A classifier *C*, which does the multi-class classification of whether a tweet is hateful, offensive, or neither.
- An adversary *D*, which uses the output of *H* to predict if a tweet is originally in AAE or not.

Instead of using a label to decide if the model's output is correct in a traditional way, the criteria for encoder *H* is whether its output "fools" the adversary *D* so that *D* can't decide if the vector outputted by *H* was original in AAE. The general structure is similar to the work of Xia et al., but I'm elaborating on what linguistic features of AAE we should eliminate. In short, I propose that we create a mapping between lexical items in AAE and non-AAE as well as checking the syntactic features commonly known in AAE. Then we perform AAE feature removal before encoding information so that AAE features are not used in the task of classifier *C*. More implementation details will be discussed in later sections.

3.2 Data

The corpora I will be using is DWMW17 (Davidson et al., 2017). The authors collected 25K tweets and classified them into *hate speech*, *offensive*, or *none*. They started with 1,000 words from HateBase, an online database of hate speech terms, as seeds. For each tweet, they crowdsourced at least 3 annotations. The data is in the following format:

count	hate speech	offensive	neither	class	tweet
3	2	1	0	0	“black bottle & a bad bitch”
3	0	0	3	2	“I love apples”

Table 3.2.1

The class label is decided by the majority vote on one of the three classifications, where 0 means hate speech, 1 means offensive, and 2 means neither, as shown in the above example. The tweets themselves are in raw format without cleaning.

This dataset does not include information of tweet author’s racial background or dialect spoken. It is extremely difficult, if not impossible, to obtain accurate information of a tweet’s author’s race/dialect background. This information can be helpful for us to reduce racial bias because it can lower false positive rate for certain racial groups given knowledge that some slurs are re-adopted by their community. Without accurate information, scholars have made attempts to generate approximate data. Blodgett et al. (2016) have done work to use the geo-location of the tweets to induce a distant supervised mapping between authors and the demographics of the neighborhood they live in (p. 1120). They drew on a set of geo-located tweets and looked up the U.S. census block group area in which tweets were sent. This way the authors were able to obtain the percentage of each race within a neighborhood as label for training. This method is a rough prediction of the demographics of where the author lives in. The model takes in a tweet and generate probabilities for each race in the following format:

ID	Tweet	AA	Hispanic	Asian	White
293846693215096832	"@ItS_niK__ I hear ya"	0.893333333333	0.0266666666667	0.0	0.08

Table 3.2.2

It is worth noticing that this prediction method is not perfect and might carry bias by itself, but having a working prediction with high accuracy is better than a random guess. I will get the percentage for AA, and if that number is higher than 0.75, set the label to be True. The reason

why I choose a higher number than 0.5 is that later the tweets will be fed to the adversary D, and if a tweet is not strongly likely to be AA/AAE to start with, it might contribute to a false sense of encoder doing well to trick adversary.

3.3 Preprocessing

DWMW17 data is in the form of csv file. I will read this data into a pandas dataframe. Then I will select the only two columns valuable for my purpose – "class" and "tweet". The tweets are in their original format and might contain a lot of noise. Therefore, it's meaningful to perform some data cleaning before moving forward.

Tweets often contain emojis. While this can communicate some information, it's too difficult to correctly interpret them without a separate set of labeled data and training. I decide to remove them in the scope of this thesis project. Other forms of text in tweets include usernames, hyperlinks, and retweets, which often don't contain meaningful information. I will remove them using a regex. The tentative regex to use is :

$$r'@[a-zA-Z0-9]+:?|[0-9]*;|https?:t.co[a-zA-Z0-9]+|[a-z]+;'$$

This regex will remove user names, hyperlinks, and emojis. I will also make all text lowercase and tokenize the words. After this step I will have a giant list of tokens each representing one word. I will split the data into 0.8/0.2, where 80% is used for training and 20% is used for testing.

3.4 Generate demographic background prediction and remove AAE features

I will use the current preprocessed data as input into the model Blodgett et al. (2016) trained to get demographic background predictions. I will add one column in my pandas

dataframe for the boolean value AA. As mentioned above, if the output from the model is higher than 0.75, AA will be set to be 1, else if will not 0.

Instead of directly translating the text input into numerical vectors, I propose to look at the text in a linguistic perspective and try to remove as many AAE features as possible so that the meaning is preserved but the lexical and syntactical features of AAE are removed so that the protected features of AAE cannot be used in the later classification tasks.

The first task is to look at lexical items that are common AAE expressions. I propose to use a dictionary that maps most common words in AAE into non-AAE and perform the swap in corpus. I propose that I will carry out a questionnaire to collect data from people who speak AAE in their daily lives. The questionnaire will consist of only one question: What are the 10 most common AAE words that you use in your life that either is not used in other dialects or has different meaning? Can you provide the corresponding translation in non-AAE? Due to the limit of scope for this thesis, I will attempt to gather this information from 15-20 participants who identify as AAE speakers. Then I can sum up the frequency for all the words that appeared in the study and choose the top 30 word mappings. This will not be a comprehensive approach that includes all AAE vocabularies, but it will be a starting point.

In addition, certain words' spelling change in AAE texting/tweeting because of their phonological variations, as proposed by Blodgett et al. (2016). They used previous studies done by Jørgensen et al. (2015) and Jones (2015) that provides a list of the most common words in AAE that involves phonological variations. Some examples of words illustrated by Blodgett et al. (2016) are shown below:

AAE	non-AAE

sholl	sure
iont	I don't
wea	where

Table 3.4.1

I will add this list of mappings into my dictionary to perform word swapping as well. An example transformation is shown from (11) to (12):

(11) sholl, I saw that n*gga yesterday

(12) sure, I saw that dude yesterday

In addition to looking at text at word level, it's also necessary to explore syntactic features. The syntactic features themselves don't seem to contribute to a false positive in hate speech detection. However, because of the use of word vectors, models are highly likely to learn the co-occurrence of these syntactic expressions with certain AAE words that are interpreted as hateful. Blodgett et al. brought attention to three common aspectual/preverbal markers: habitual *be*, future *gone*, and completive *done.* Example of each of the phenomenon is shown below:

(13) I be scared all the time around him. – habitual *be*

(14) Then she gone be single Af. – future *gone* (Blodgett et al, 2016)

(15) He done talking about his work. – completive *done*

In order to search for such constructions, we need to search for the keywords as well as the syntactic environment around the keyword. To reach this goal, POS (part of speech) tagging is necessary. Python nltk module has pos_tag() function that can perform this function. Blodgett et al. proposed a better POS tagging tool – the ARK Twitter POS tagger (Gimpel et al., 2011; Owoputi et al., 2013). This is particularly useful in our case because Jørgensen et al. (2015) have shown that it produces similar accuracy rates on both AAE and non-AAE tweets. Then we can

go through the list of words and look for constructions O-*b/be*-V, *gone/gon/gne*-V and *done/dne*-V, as proposed by Blodgett et al (2016). I propose that when I do locate habitual *be,* I will look at the object I found before and conjugate accordingly. Similarly, when I locate future *gone,* I will check the object proceeding it and add the respective conjugation of *be* before it and change *gone* to be *going to.* As for completive *done,* I will do the same except I will add the corresponding form of have before it. (16) to (18) are corresponding outputs from (13) to (15):

(16) I am scared all the time around him.

(17) Then she is going to be single Af.

(18) He has done talking about his work.

Another syntactic phenomenon common in AAE is the omit of be (and its variant) before nouns or adjectives. Here are two examples:

(19) Malcolm, he kinda big. – Before adjective

(20) You the one messed up, not them. – Before noun

After running the POS tagger as mentioned above, I can conduct a sequential search on sentences without a verb with the structure N-O or N-adj and perform insertion of conjugated be. After such operation (19) and (20) will be transformed into (21) and (22):

(21) Malcolm, he is kinda big.

(22) You are the one messed up, not them

Multiple negation is also a prominent feature of AAE. This can be in the form of double negation or even three. The use of “ain’t” is a common practice.

(23) You ain't nothing to me.

(24) Don't nobody say nothing to her.

I propose to use a set of words signifying negation without "not" – *nobody, no one, nothing*. Then we search for multiple negation keywords that are used together. If we see "ain't", change it to be *be* + *not* conjugated according to the subject. We keep the negation with "not" but change other negation to their existential form: *nobody-anybody, no one- anyone, nothing -anything*. Although this does not capture all instances of multiple negation, it serves to locate the basic patterns. (23) and (24) will be (25) and (26), respectively:

(25) You are not anything to me.

(26) Don't anybody say anything to her.

I want to emphasize that even though these expressions don't seem to be offensive or hateful themselves, models might learn the correlation between these features and other vocabulary often used in AAE that can be misjudged as hateful, thus contributing to higher false positive rate.

3.5 Encoder H

Machine learning models cannot process text directly, and they have to be transformed into real numbers of certain forms. A common practice is to encode words into vectors, where the vectors in space represent semantic meaning of words, known as word embeddings. The distance between vectors corresponds to semantic relatedness of words as well. For example, if we encode words into 2-dimensional vectors, and we have the word "water" with vector (2, 0), "ice" with vector (2, 1), and dog (5, 6). We can tell that "water" and "ice" have vectors closer to

each other than dog because they are more related semantically. Following the approach of Sap et al. (2019), I will initialize the model with GloVe vectors (Pennington et al., 2014). They are preferred because they have the ability to retain both local and long-distance word co-occurrences. This is easy to understand since a given word is often not only affected by words immediately surrounding them. To construct GloVe vectors, we will create a symmetric matrix that records the co-occurrence of tokens. Take an example sentence:

(27) The dog barks at the door.

	the	dog	barks	at	door
the	0	1	0	1	1
dog	1	0	1	0	0
barks	0	1	0	1	0
at	1	0	1	0	0
door	1	0	0	0	0

Table 3.5.1

Then we analyze the co-occurrence relationship of every 3 tokens to obtain probability ratio.

After GloVe initialization, I will use BiLSTM with attention mechanism, as proposed by Sap et al. (2019) and Xia et al. (2017). BiLSTM refers to bidirectional LSTM (long short term memory). The advantage it has over one directional LSTM is that they can capture information of occurrence in both directions. LSTM is a type of RNN (Recurrent Neural Network) that is designed to support sequence like input data, such as corpus input or time series. LSTM is known to be able to learn the complicated dynamics of ordering of input sequences, which is a promising feature for natural language processing.

My goal is to generate encodings of text that confuses the adversary D , which attempts to make predictions if the original text is in AAE. To work towards this goal, I will train the BiLSTM so as to minimize the cross-entropy of a text input's annotated AAE label and the prediction given by adversary D. This is similar to the process in Sap et al. (2019, p. 1670). For two probability distributions, cross-entropy measures the amount of information needed to identify one event from one distribution given the other. Cross-entropy measures the similarity between two distributions, and therefore minimizing it in my setting equals to eliminating connection between AAE label and adversary prediction. This means that we are trying to remove the protected features of AAE as much as possible in the encoding process. In some output vectors, if one data is extremely different from the rest, I propose to drop it because it will likely correspond to situations like (2) and (3).

3.6 Adversary D and Classifier C

I'm including these two classifiers here together because they will be very similar. They both take in an encoded high dimension vector and perform classification tasks – adversary D tried to decide if the original text is AAE, and classifier C predicts if the original text is hate speech.

Davidson et al. (2017) established that linear SVM and Logistic Regression with L2 Regularization work the best for hate speech classification tasks. On the other hand, Xia et al. (2017) proposes to use two-layer MLP with *tanh* activation function. Logistic regression works by trying to draw a hyperplane between data clusters to obtain classification. Regularization is used to prevent the model from overfitting. L2 regularization uses the mean squared error as the regularization term. The model then tries to minimize both the lost function and the regularization term. SVM stands for support vector machine, and it works by trying to draw a

hyperplane between data clusters as well. Support vector is the data closest to the hyperplane, and we try to maximize the perpendicular distance between support vector and hyperplane. One point to notice is that this is not necessarily two-dimensional – it can be multi-dimensional, and the hyperplane is always one dimension lower than the space we are analyzing in. MLP stands for multi-layer perceptron, which is a type of feedforward neuron network. It consists of three types of layers- input layer, hidden layer, and output layer. The neurons in hidden layers are trained with back propagation learning algorithm (Abirami and Chitra, 2020). MLP is specifically designed to approximate continuous functions that can't be separated linearly. The computation in each neuron, as described by Abirami and Chitra, is below:

$$O(x) = G\big(b(2) + W(2)h(x)\big) \tag{4}$$

$$h(x) = s(b(1) + W(1)x) \tag{5}$$

b(1) and b(2) are bias, W(1) and W(2) are weight matrices, and s and G are activation functions, which in my case is *tanh:*

$$\tanh(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}} \tag{6}$$

I plan to experiment with these three methods and adopt the one with the best performance.

4 Result Analysis

To evaluate my model, I have two main aspects I need to investigate. How does the model perform in terms of hate speech detection? Does the model reduce racial bias against AAE users? If so, by how much? In addition, I also want to evaluate how much improvement the AAE related feature removal helped on racial bias reduction.

4.1 Hate Speech Detection Analysis

Firstly, I will perform analysis on hate speech detection. I will use the work of Xia et al. as the baseline (2017). The statistics I will use for this part are accuracy and F1 score. Accuracy refers to the fraction of time that my model correctly predicts the class label. The reason why I include F1 here in addition to accuracy is that accuracy is not a good evaluation metrics in certain situations, such as when there is imbalance of class in data. Class imbalance is highly likely in my case because the majority of Twitter data will not be hateful. This means that if my model just predicts everything to be not hate speech the accuracy will still look promising, contributing to false sense of success in the task. F1 score takes into consideration both precision and recall, where precision measures how many predictions of positives are correct and recall measures how many positives the model is able to find of all true positives. I will then compare this data with the baseline model. The table will be in the following format:

	Accuracy	F1
baseline	90.68	76.05
ours	?	?

Table 4.1

In addition, I will analyze how accuracy and false positive rate change over training epochs, following Xia et al. (2017). This will inform me how false positive rate and accuracy trade off during the process of training.

4.2 Racial Bias Reduction Analysis

I hope to evaluate the effectiveness of both models with and without AAE feature removal, and compare with the baseline model. Because I planned to apply the racial background prediction model Blodgett et al. (2016) on my corpus, I will have access to race labels. I will

compute the false positive rate for both AAE and non-AAE texts and compare the ratio. The table summary will be in the following format:

	fpr(AAE)/fpr(non-AAE)
baseline	
ours (no feature removal)	
ours	

Table 4.2

I will also test if the racial bias in my model's prediction result is still statistically signicant.

5 Future Work

To further improve on this project design, I can apply the model trained on Davidson et al. (2017) to a different dataset to test if the model can generalize well and does not overfit. A potential candidate is Waseem and Hovy (2016). This dataset labeled tweets as "Racism", "Sexism", and "Neither". To make our model applicable, I will need to perform some processing on the data and combine "Sexism" and "Racism" to be a single label.

Xia et al. (2017) also points out that it has been shown that multi-task learning on similar tasks can shift focus to toxicity related elements in hate speech detection. One example would be racial identify prediction. Instead of using a identity prediction as adversary, we can incorporate it as the second task in a multi-task learning model.

6 Conclusion

In this thesis I have explored the issue of racial bias against AAE speakers for hate speech detection models on social media. Existing approaches to address the issue includes racial and dialectal priming on data level (Sap et al., 2019), smoothing for phrases with high correlation with certain classes (Mozafari et al., 2020), and adversarial training to remove dialectal information(Xia et al., 2017), etc. There is a big gap between how humans perceive offensiveness and hate through language and automatic hate speech detection models. Babou-Sekkal (2012) argues that people who use the same language does not necessarily share the same set of socio-linguistic rules. Therefore they might use different dialectal expressions that might confuse each other. I proposed an approach that processes text and remove lexical and syntactical features of AAE as much as possible before sending text to be encoded into high dimension vectors. Adversarial training is then used to demote the model from using AAE related information to make hate speech detection. By removing AAE features, I am reducing the gap between the two dialects and making data created by the two groups of authors share similar expressions, and potentially socio-linguistic rules, as mentioned by Babou-Sekkal. However, there still remains many aspects where machine learning models are lacking when understanding human offensiveness. It would be beneficial for us to explore how models can detect socio-linguistic cues, such as euphemism and insider humor, as well as semantic information of entailment and presupposition and how syntactic scoping can affect offensiveness of sentences.

References

Abirami,S., Chitra, P. (2020). Chapter Fourteen - Energy-efficient edge based real-time healthcare support system. Editor(s): Pethuru Raj, Preetha Evangeline. Advances in Computers, Elsevier,Volume 117, Issue 1, 2020. Pages 339-368. https://doi.org/10.1016/bs.adcom.2019.09.007.

Anderson, Luvell & Lepore, Ernie. (2013). Slurring Words. Noûs 47 (1):25-48.

Babou-Sekkal, Meryem. (2012). A Sociolinguistic Analysis of Use and Perception of Insults: Tlemcen Speech Community. Dissertation, University of Tlemcen.

Blodgett, Su Lin, Green, Lisa & O'Connor, Brendan. (2016). Demographic Dialectal Variation in Social Media: A Case Study of African-American English. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1119–1130, Austin, Texas, November 1-5, 2016. Association for Computational Linguistics

Camp, Elisabeth (2018). A dual act analysis of slurs. *Oxford Scholarship Online*. https://doi.org/10.1093/oso/9780198758655.003.0003

Davidson, Thomas, Warmsley, Dana, Macy, Michael, & Weber, Ingmar. (2017). Automated hate speech detection and the problem of offensive language. arXiv:1703.040 https://arxiv.org/pdf/1703.04009.pdf

Davidson, Thomas, Bhattacharya, Debasmita & Weber, Ingmar. Racial Bias in Hate Speech and Abusive Language Detection Datasets. (2019). Proceedings of the Third Workshop on Abusive Language Online, pages 25–35. Association for Computational Linguistics.

Eisenstein, J., Ahmed, A., and Xing, E. P. (2011). Sparse Additive Generative Models of Text. In ICML'11.

ElSherief, Mai, Kulkarni, Vivek, Nguyen, Dana, Wang, William Yang, & Belding, Elizabeth. (2018). Hate Lingo: A Target-Based Linguistic Analysis of Hate Speech in Social Media. Twelfth International AAAI Conference on Web and Social Media. https://www.aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/viewFile/17910/16995

Founta, Antigoni-Maria, Djouvas, Constantinos, Chatzakou, Despoina, Leontiadis, Ilias, Blackburn, Jeremy, Stringhini, Gianluca, Vakali, Athena, Sirivianos, Michael & Kourtellis, Nicolas. (2018). Large scale crowdsourcing and characterization of twitter abusive behavior. In ICWSM.

Gimpel, Kevin, Schneider, Nathan, O'Connor , Brendan, Das, Dipanjan, Mills, Daniel, Eisenstein, Jacob, Heilman, Michael, Yogatama, Dani, Flanigan, Jeff & Smith, Noah A. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In Proceedings of the 49th

Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 42–47. Association for Computational Linguistics, 2011.

Jones, Taylor. Toward a description of African American Vernacular English dialect regions using “Black Twitter”. American Speech, 90(4): 403–440, 2015.

Jørgensen, Anna Katrine, Hovy, Dirk & Søgaard, Anders. Challenges of studying and processing dialects in social media. In Proceedings of the Workshop on Noisy User-generated Text, pages 9–18, 2015.

Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). Hate speech detection and racial bias mitigation in social media based on Bert Model. PLOS ONE, 15(8). https://doi.org/10.1371/journal.pone.0237861

Owoputi, Olutobi, O’Connor, Brendan, Dyer, Chris, Gimpel, Kevin, Schneider, Nathan & Smith, Noah A. Improved part-of-speech tagging for online conversational text with word clusters. In Proceedings of NAACL, 2013.

Pennebaker, J. W., Boyd, R. L., Jordan, K. & Blackburn, K. (2015). The Development and Psychometric Properties of LIWC2015. https://goo.gl/1n7y5A.

Pennington, Jeffrey, Socher, Richard & Manning, Christopher D.. (2014). GloVe: Global vectors for word representation. In EMNLP.

Sap, M., Card, D., Gabriel, S., Choi, Y., & Smith, N. A. (2019). The Risk of Racial Bias in Hate Speech Detection. ACL. pp. 1668-1678. https://homes.cs.washington.edu/~msap/pdfs/sap2019risk.pdf

Waseem Z, Hovy D. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In: Proceedings of the NAACL Student Research Workshop. San Diego, California: Association for Computational Linguistics; 2016. p. 88–93.

Xia, Mengzhou, Field, Anjalie & Tsvetkov, Yulia. (2017). Demoting Racial Bias in Hate Speech Detection. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 7–14. Association for Computational Linguistics.

https://aclanthology.org/2020.socialnlp-1.2.pdf