

Detecting and Reducing Bias in a High Stakes Domain

Ruiqi Zhong¹*, Yanda Chen¹, Desmond Patton², Charlotte Selous², Kathleen McKeown¹

¹Department of Computer Science, Columbia University

²School of Social Work, Columbia University

{rz2383, yc3384, dp2787, cis2114}@columbia.edu
{kathy}@cs.columbia.edu

Abstract

Gang-involved youth in cities such as Chicago sometimes post on social media to express their aggression towards rival gangs and previous research has demonstrated that a deep learning approach can predict aggression and loss in posts. To address the possibility of bias in this sensitive application, we developed an approach to systematically interpret the state of the art model. We found, surprisingly, that it frequently bases its predictions on stop words such as “a” or “on”, an approach that could harm social media users who have no aggressive intentions. To tackle this bias, domain experts annotated the rationales, highlighting words that explain why a tweet is labeled as “aggression”. These new annotations enable us to quantitatively measure how justified the model predictions are, and build models that drastically reduce bias. Our study shows that in high stake scenarios, accuracy alone cannot guarantee a good system and we need new evaluation methods.

1 Introduction

There has been increased interest in using natural language processing to help understand the root causes of violence in cities such as Chicago. Researchers in social work found that posts on social media by gang-involved youth mirror their life on the physical street (Patton et al., 2016, 2017) and suggested that if community outreach workers could use tools to help identify posts of aggression and grief, they could intervene before grief turns to retribution and help avert violence. Early research in this area (Blevins et al., 2016) used support vector machines for the three-way classification task of *aggression*, *loss* and *other* and recent work (Chang et al., 2018) demonstrated a

significant increase in accuracy through the use of a neural net approach along with the representation of contextual information from previous conversations. In both of these approaches, computer scientists and social work researchers worked side by side to enable qualitative analysis of the data to use in machine learning and to provide meaningful error analyses.

Given the interest in using this work in a live setting to identify situations where community outreach workers could intervene, we are concerned with *how* the model makes the decision. This is especially important since this is a high stakes domain and machine learning systems are known to incorporate bias. For example, the COMPAS Recidivism Algorithm, which assesses the likelihood of a criminal defendant to re-offend and influences judges’ sentencing decisions, is known to be biased against African Americans (Feller et al., 2016). To avoid similar errors in the gang-violence application, we examined whether the model is biased through systematic investigation of how the model makes the predictions. Since misclassifying a tweet as *aggression* might potentially criminalize a vulnerable population (young black and latino youth) that is already over criminalized, we focus on classification of the *aggression* class in our paper.

Our finding is surprising: in at least 10% of the cases the Chang et al. (2018) model relies on stop words such as “a” and “on” to classify a tweet as aggression; these words carry no semantic or pragmatic indication of aggression or loss. We recognized that there might be more hidden biases, so we developed novel metrics to quantitatively compare the difference between how humans and models make their classification. To achieve this, we ask domain experts to annotate the rationale for the aggression tweets by highlighting words and phrases that indicate why experts classify them as

*The first author is now a PhD student at University of California, Berkeley.

aggression. We compare the rationale against the models’ interpretation.

We also develop a model to incorporate human rationales, and show that the system can make predictions based on similar reasons to those of domain experts. Our F-score is slightly improved over [Chang et al. \(2018\)](#) and we are able to reduce the bias drastically. We show how important this is by using an adversarial approach which demonstrates that we can change the prediction of posts predicted as *other* to *aggression* by the simple insertion of the non-offending function words. This implies that high F-score alone does not guarantee a good system, and that researchers in other high stakes domain should evaluate their algorithms from various angles.

Our contributions include:

- A validated finding that the previous state-of-the-art model is biased.
- A rich dataset created by domain experts containing annotated human rationales.
- New metrics to evaluate models’ robustness under adversary, and the difference between model interpretations and human rationales.
- A model that incorporates human rationales in training and behaves more as humans do.

We also include an in-depth discussion on how misinterpretation might emerge when labels are being annotated by the experts. The battle against bias is far from solved, and we hope our insights and methodologies can guide future research in high stakes domains.

2 Revealing Model Biases

We used the code and data from [Chang et al. \(2018\)](#) and fully reproduced the state-of-the-art model that uses a Convolutional Neural Network, domain specific word embeddings and context features.

2.1 Locating Biases

We examine the model using the simple leave-one-out method similar to [Ribeiro et al. \(2016\)](#). Suppose each tweet T is a sequence of l tokens $[e_1, e_2, \dots, e_l]$ and the model gives the prediction confidence score y . For each token e_i , we calculate the model’s prediction confidence y_{-i} on the same tweet with e_i masked as an unknown word; hence

the influence of the i^{th} token I_i is approximated by $I_i = y - y_{-i}$. Then we manually examine the top two influential uni-grams for each tweet that the model classified as aggression.

As we expect, tokens strongly associated with aggression, such as “”, “dead” and many other curse words appear frequently as the most influential tokens. However, surprisingly, words such as “a” and “on” are frequently considered the most influential by the model as well. Out of 419 tweets that the model classifies as aggression, 15/32 times the model considers the token “a” to be the top first or second most influential token and “on” 24/51 times the top first or second, respectively. We also found that the model is biased against other stop words, such as “into” and “of”. This indicates that more than **10%** of the time the model heavily relies on these unintentional differences of language use to classify tweets as aggression!

2.2 Confirming Biases Exist

Initially, we found this unexpected result surprising, so we checked the second order sensitivity ([Feng et al., 2018](#)) and constructed adversarial data sets.

Second Order Consistency [Feng et al. \(2018\)](#) defined this term when they found that the leave-one(token)-out (LOO) method might find a completely different set of most influential words if the most un-influential word is dropped from the sentence. However, we do not observe this problem in our case: out of 419 predictions, 379 times the most influential word remains the most influential and 33 times it becomes the second most influential one after we drop the most un-influential words.

Adversarial Dataset If we can identify a weakness of the model, we must be able to break it in a corresponding way. To manifest that there is indeed bias against “a” and “on,” for each of these two words, we inserted it in each labeled tweet and counted the number of times the model changes its prediction from non-aggression to aggression.

To ensure that our insertions are natural, we trained a large language model ([Peters et al., 2018](#)) on the unlabeled corpus (~ 1 million tweets) used by [Chang et al. \(2018\)](#), and score the naturalness of the insertion by the change of perplexity given by the language model. For each tweet we calculate the most “natural” position to insert our biased word and only took the top 800 natural

tweets out of ~ 7000 candidates as our adversarial dataset. We also subsample this dataset and check the subsample manually to make sure that these insertions make sense. In fact, since most tweets in the dataset are grammatically flexible in the first place, our insertion methods usually do not make them more unnatural - we have fewer constraints than adversarial attacks (Alzantot et al., 2018) conducted in domains such as IMDB reviews (Maas et al., 2011) or Textual Entailment (Bowman et al., 2015).

We find that even with these naive and seemingly harmless insertions, the model changes the prediction from non-aggression to aggression for about 3%/5% of the generated tweets when inserting “a”/“on”. With this overwhelming evidence, we conclude that the model is biased.

The simplest way to fix this specific bias is a post-mortem one - just remove “a” and “on” in training and prediction. However, this only works if these are the only two points of bias and we suspect that there are many more biases lurking in the model. Hence, we need to comprehensively study the difference between how humans and models make their decision. We need a new dataset with annotated human rationales to achieve this goal.

3 Data

We use the same dataset and cross validation folds as in Chang et al. (2018), which contains 329 Aggression tweets, 734 Loss tweets, with the Other class comprising the remaining 3,873 tweets. The language in the dataset is hyper-local to the community we are studying and differs significantly from standard American English. Therefore, most of the toolkits (e.g. pre-trained language models, parsers, etc) developed for standard English do not apply. We extend their dataset by asking domain experts to annotate the rationales and additional information on context for each of the tweets labeled as aggression. Below is the set of annotation instructions.

- **Rationales in Tweets** We ask the domain experts to annotate the most important word/words (rationales) in the tweet that makes them think the tweet is aggression/loss; or say, the removal of which would make them not consider it as an aggressive tweet. We ask them to annotate the “minimum set of words” that is important for their final decision, thus excluding words that are

not very meaningful, such as “on”, “basically”, “lol”, etc. Examples are shown in Table 1.

- **Aggressive by Context** Sometimes the tweet itself does not contain aggressive content if we interpret them at face level, but placing it into context reveals that the user in fact conveyed aggressive intent. We ask the experts to label this attribute as True whenever they use an outside source to provide additional information needed to label a tweet. These sources can include news sources, past posts, interactions, online dictionaries, or reddit threads.
- **Aggressive by Mention/Retweet** Similarly, interactions between users that are suspected to be affiliated with the same or rival gangs might contain aggressive intent. For example, users may communicate with others in the same gang about the possibility of retaliation or they may directly taunt members of rival gangs. We ask the annotators to label this attribute as True if the context of interaction between users contributes to the aggression label. This corresponds to the pairwise interaction feature in Chang et al. (2018).
- **Aggressive by Picture/URL** This attribute indicates whether the picture or URL in the tweet contains aggressive content. Since our study is natural language processing based, we do not make use of this piece of information in this paper.
- **Controversial Labels** Occasionally our experts would interpret a tweet previously considered aggressive as non-aggressive. We do not use these labels in our experiments for fair comparison with Chang et al. (2018), but we include them in our new dataset.

Out of the 329 aggression tweets, 274 contain rationales in the tweet, while the other aggression labels are inferred by context rather than the content within the tweet itself. To keep notations uncluttered, we refer to all the “aggressive by *” labels as *context labels* in the rest of this paper. The meta-statistics on this can be seen in table 2. Additionally, 39 (12% of aggression) tweets are considered controversial by our experts. We also ask the annotator to write down thoughts and observa-

Tweets
“@user_ do you think it’s cool yo ass slow ? You sound <u>dumb asf</u> ”
“Rob you fucking <u>goofys</u> that got some to say about <u>rage</u> just to show y’all <u>bitches</u> ”
“GlizzyGang Bitch We Got Out <u>Glocks</u> Up🐱”

Table 1: Typical aggression tweets in our dataset, which includes annotations with human rationales (underlined).

tions while annotating, and we share these insights in our discussion section.¹

Context Label Types	Yes	No	N/A
Context	107	74	148
Mention/Retweet	36	35	258
URL	9	3	317
Picture	14	15	300

Table 2: Whether aggressive intent is observed in Context, Mention/Retweet, URL and/or Picture. Many of the context label cannot be annotated due to Twitter access policy.

4 Methods

We compare the Convolutional Neural Network (CNN) model from Chang et al. (2018), an LSTM (Long Short Term Memory) Network model with attention mechanism (architecture defined below) and our new model that incorporates rationales by trained attention (Zhong et al., 2019).²

4.1 Overview of Chang et al. (2018)

Here we give a quick overview of their CNN model with context features. Let l denote the length of the pre-processed tweet, $T = [e_1, e_2 \dots e_l]$ be the input sequence, and c the context feature. First each token e_i is mapped to an embedded representation w_i through their domain specific word embeddings. Then they used a convolution layer with kernel size 1 and 2, followed by a maxpooling layer and a linear fully connected layer to produce the tweet representation z . Finally, they concatenated z with the context feature c and fed it to the

¹Due to privacy concerns we cannot release the additional annotation publicly, but we will make it available to researchers who sign an MOU specifying their intended use of the data and their agreement with our ethical guidelines. Send email to the dp2787@columbia.edu. Please contact the authors of Chang et al. (2018) for the original dataset.

² Our code is publicly available at https://github.com/David3384/GI_2019

final classification layer. We refer the readers to their original paper for further details.

4.2 LSTM with Attention

We use a standard LSTM model with attention mechanism. The model encodes the embedded sequence $[w_1, w_2, \dots, w_l]$ with an LSTM to obtain hidden state representation h_i with hidden dimension 64 for each position i and then calculate an attention weight for each position based on h_i . The final representation z is a weighted average of the hidden states concatenated with the context feature, which we pass to the final classification layer. Formally,

$$\alpha_i = \tanh(\mathbf{v}h_i), A(i) = \frac{e^{\alpha_i}}{\sum_j e^{\alpha_j}} \quad (1)$$

$$z = [\sum_{i=1}^l A(i)h_i, c] \quad (2)$$

where \mathbf{v} are trainable weights we used to calculate the attention. We used binary cross entropy loss between the final prediction and the ground truth label, which we denote as \mathcal{L}_{clf}

4.3 Trained Attention with Rationale

We guide the model attention to attend to human rationales. We introduce a “ground truth attention” A^* , which is uniformly distributed over all the positions where the token is an annotated rationale. For example, if $R = \{r_1, r_2 \dots r_d\}$ are indexes of the tokens that are rationale words, then the ground truth attention distribution A^* is defined as:

$$A^*(i) = \frac{1}{d} \text{ if } i \in R, 0 \text{ otherwise} \quad (3)$$

Then we impose a KL divergence loss component between the ground truth attention and the model’s attention. We optimize the weighted sum between the label classification and the attention loss, i.e.

$$\mathcal{L}_{attn} = KL(A^*||A), \mathcal{L} = \mathcal{L}_{clf} + \lambda_{attn}\mathcal{L}_{attn} \quad (4)$$

where we fix $\lambda_{attn} = 4$ after tuning.

To make a fair comparison with Chang et al. (2018), we use the same set of hyper-parameters and training procedures. We find that taking the majority vote of an ensemble of 5 neural networks trained on the same data can improve the F-score by 1~2 points, so we report both the averaged and the majority vote result across 5 independent

5 Evaluation Metrics

5.1 Rationale Rank

We further relax our requirement for the model: we do not have the unrealistic hope that the model is relying on every rationale word; instead, we only consider the lowest ranking rationale. Formally, suppose there are multiple rationale words with indexes $R = \{r_1, r_2..r_d\}$, then we define the “rationale rank” as:

$$rationalerank = \min_{i \in R} rank(i) \quad (5)$$

Since the majority of tweets usually has rationale rank 0, the averaged rationale rank is small and the difference between each model is “diluted” by the prevailing zeros. Therefore, for each model, we also report the percentage of tweets that has rationale 0 or 1, respectively. This more intuitively reflects on what fraction of the tweets the models’ reasoning agrees with human rationales.

Chang et al. (2018) claimed that their pair-wise interaction feature captures information about users’ past interactions, and they supported it by showing significant improvement when incorporating it into the CNN model. Nevertheless, it does not automatically entail that the model is using this feature in a way that humans do. Here we examine this claim quantitatively with our annotated interpretation of context.

$$y = \sigma(w_z z + \sum_{i=1}^3 w_i c_i + b) \quad (6)$$

We group the tweets by the annotated context label “by mention/retweet”, a single label, annotated as described in section 3. If the pair-wise interaction feature c_3 truly models domain experts’ interpretation of user interaction as Chang et al. (2018) has hypothesized, $w_3 c_3$ should be higher for those tweets that are annotated as “aggressive by mention/retweet” than those that are not. We do not examine use of other context labels and features, since other context features do not model URL, pictures or a general type of context.

6 Prediction Flips under Adversary

We consider the simplest operation: inserting certain candidate neutral unigrams $u \in U = \{\text{“a”}, \text{“on”}, \text{“da”}, \text{“into”}, \text{“of”}, \text{“that”}\}$ that the model strongly associates with aggression.³ Assuming that we have already trained a language model that gives a perplexity score $p(T)$ for a tweet $T = [e_1, e_2, \dots, e_l]$, then for each $u \in U$ we create a

³We describe how we choose these unigrams in the appendix A.2. We also experimented with “in”, “be”, “do”, “any”, “u”, “out”, “basically”, “yea”, “ever”, “u”, “”, “we” etc, but the results are not significant so we present them in the appendix.

dataset by inserting it into all of the tweets T in our labeled dataset \mathcal{T} . We describe how we train the language model in the appendix section A.1.

For each $T \in \mathcal{T}$, there are $|T| + 1$ positions to insert the unigram u . We pick the resulting tweet T' with u inserted that has the highest likelihood score $p(T')$ among all the $|T| + 1$ possible insertions, and calculate the corresponding “insertion naturalness score” $p(T') - p(T)$. After inserting the unigram u into all the tweets in the labeled dataset \mathcal{T} , we obtain a corresponding dataset \mathcal{T}' and pick the top 800 tweets that have the highest “insertion naturalness score”.

Then we calculate how many times a model would flip its prediction from non-aggression to aggression among these 800 pairs of tweets, before and after inserting the unigram u . Notice that even one percent of “flip rate” is significant: the unigrams u we consider here are not rare words and thus have high coverage. If the model is deployed at a large scale, millions of tweet will be classified, and tens of thousands of tweets will be influenced, not mentioning that we are considering only one unigram and there are many potential other tokens biased by the model.

6.2 Evaluating Generation Naturalness

Since words are discrete, generating adversarial text inputs has been a hard problem in other domains for at least two reasons: 1) one perturbation might be large enough to change the label. 2) we cannot guarantee that the resulting text is still a natural input. However, these challenges do not play much of a role in our domain. First, the candidate words we are inserting are “function words”, which we know should not cause the model to flip their predictions. Second, the large proportion of tweets we are working with are grammatically flexible in the first place. Therefore, in contrast to genres that follow the conventions of Standard American English grammar, we suspect that adding function words will not make the tweets look unnatural to the domain experts and we test this hypothesis.

To evaluate the plausibility of our generated tweets, we sample 18 tweets from all the generated tweets and 18 existing tweets from our labeled dataset. Then we ask our experts to try their best to classify whether each tweet comes from model generation or the existing dataset. We also ask them to annotate whether each tweet is “natu-

ral”, “reasonable but awkward” or “nonsensical”.

The experts classified 25 of them correctly ($\sim 70\%$), only slightly better than the 50% random guess baseline even though they are aware that we are generating tweets by inserting function words. Out of 18 generated tweets, they considered 7 as “natural”, 8 “reasonable but awkward” and 3 as “non-sensical”. In comparison, out of the 18 existing tweets, they considered 14 as “natural”, 1 as “reasonable but awkward” and 3 as “non-sensical.” These results imply that although our generated tweets might not all be fluent, they are still reasonable enough as adversarial tweets. Interestingly, they also considered 3 existing tweets as “non-sensical”, partly because the tweets are highly grammatically flexible and we told them that there are generated tweets.

7 Results

The model F-score performances are shown in table 3 and rationale rank in table 4. We present our results for macro F-score, Rationale Rank, Prediction Flips on Adversarial Dataset and Context Feature Evaluation. We also evaluate the models from (Blevins et al., 2016) for completeness.⁴

F-score We first notice that taking the ensemble of 5 independent runs trained on the same data consistently improves the performance over averaged individual run performance, so we report all of our results on ensemble models. Compared to Chang et al. (2018), our LSTM+Rationale model differs by at most one point on all of the macro-F score metrics, and sometimes achieves the best performance.

Rationale Rank A closer look at the rationale rank result reveals a huge difference (p-value $< 10^{-5}$): our model significantly beats all other models on this metrics.

Prediction Flips under Adversary The results are shown in table 5. For each candidate unigram we count the number of times the model flips its prediction from non-aggression to aggression among the top 800 insertions. By incorporating rationales into the model, we achieve the fewest flips for half of the unigrams and under none of the six cases does it flip the most predictions.

These two results demonstrate that although the CNN model (Chang et al., 2018) only slightly un-

⁴For consistency, we used their model but first classify aggression vs. rest, rather than other vs. rest as in (Blevins et al., 2016). The training procedure is deterministic, so the ensemble gives the same performance as individual ones.

Model	Average Macro F1	Ensemble Agg. F1	Ensemble Loss F1	Ensemble Other F1	Ensemble Macro F1
Blevins et al. (2016)	63.5	34.5	67.3	88.7	63.5
Chang et al. (2018)	68.2	41.3	75.4	91.3	69.4
CNN + Twitter	65.8	36.2	73.9	91.3	67.1
LSTM	67.0	38.4	74.2	91.3	68.0
LSTM + Rationale	68.2	43.0	75.0	91.6	69.8

Table 3: Results comparing different model architectures. Chang et al. (2018) refers to the full CNN model with domain specific embedding and context feature; “CNN Twitter” replaces the domain specific embedding with GloVe twitter embeddings (Pennington et al., 2014); LSTM refers to LSTM Attention described in section 4.2; LSTM Rationale refers to the same LSTM Attention with trained attention described in section 4.3. For each column, the best performing entry is bolded.

Model	Avg RR	RR = 0	RR = 1
Blevins et al. (2016)	1.70	0.60	0.13
Chang et al. (2018)	1.42	0.54	0.17
CNN + Twitter	1.82	0.43	0.25
LSTM	1.73	0.50	0.15
LSTM + Rationale	0.86	0.69	0.14

Table 4: Results on (averaged) rationale rank (RR), fraction of tweets when RR = 0 and RR = 1, respectively. For (averaged) rationale rank and RR = 0, the best performing entry is bolded. We do not bold the column RR = 1 since it is sub-optimal.

derperforms our model in terms of F-score performance, it behaves very differently behind the scene. Our model is more robust and makes predictions that match human predictions. This implies that traditional F-score measure alone neither provides us deeper understanding of the model, nor reveals the underlying biases.

Context Feature Evaluation We calculate the correlation between “how the human uses context” and “how the model uses context” as described in section 5.2. The mean difference of w_3c_3 between the two groups split by the label value is 1.61 with statistical significance 10^{-3} , indicating that the pairwise interaction feature correlates well with how domain experts interpret users’ retweet/mentions.

8 Related Work

Patton et al. (2013) initiated a new field of study, coining the term *internet banging* to describe the phenomenon whereby online altercations lead to offline violence. This work is further motivated by challenging gun violence statistics in the city of Chicago which saw a 58% spike between 2015 and

2016. In addition, the number of homicides stemming from physical altercations decreased during that period with little empirical evidence to explain this shift (Kapustin et al., 2016). Investigating internet banging represents an important first step in understanding why gun deaths increase despite fewer physical altercations.

Blevins et al. (2016) is a collaboration between social work and computer science experts that brings together social work’s substantive knowledge of communities, youth development, and gun violence with data science’s innovative machine learning tools. They developed a new analytical system that leverages community participatory methods and machine learning to automatically detect the expression of loss and aggression on Twitter. Blevins et al. (2016) collected and annotated tweets posted by Gakirah Barnes, a deceased gang members prolific on twitter, as well as her top communicators. Chang et al. (2018) extended this dataset to be six times larger and achieved the state-of-the-art result by using a CNN.

Our interpretability method is most similar to the leave-one-out (LOO) approach by Ribeiro et al. (2016). Nevertheless, it risks making the sentence ungrammatical, hence producing “rubbish examples” (Goodfellow et al., 2014) and pathological interpretations (Feng et al., 2018). We explicitly address this concern by checking the second-order consistency proposed in Feng et al. (2018). To further ensure that the removal of a word results in a “blank state”, erasure-based method proposed in Li et al. (2016) can be used. It would also be interesting to explore different interpretability methods, such as using the influence function (Koh and Liang, 2017) or nearest neighbor in the feature space (Strobelt et al., 2019). Our method of automatically generating adversarial data accord-

Models \ Unigrams	a	on	da	into	of	that
Blevins et al. (2016)	44.16 [!]	121.00 [!]	37.44 [!]	117.72 [!]	36.84 [!]	74.48 [!]
Chang et al. (2018)	26.56	36.00	8.52 [*]	21.68	19.20	7.52 [*]
CNN + Twitter	38.40	35.80	17.60	15.88	5.24 [*]	7.64
LSTM	16.16	40.28	21.52	26.92	13.20	13.12
LSTM + Rationale	13.52 [*]	23.16 [*]	12.40	10.28 [*]	11.40	8.96

Table 5: The number of model’s prediction flip from non-aggressive to aggressive, out of the 800 attacking tweets generated by inserting a specified “neutral” unigram. To obtain a stable estimate, the count of number of flips of each model is averaged across 5 model runs and models trained on 5 folds - thus leading to non-integer results. For each column, the worst performing entry is marked with “!” and best with “*”.

ing to a pre-identified weakness of a model without accessing the model output and parameters is most similar to Jia and Liang (2017). From Alzantot et al. (2018), we borrowed the idea of using a language model to ensure naturalness of generated sentences, and it is a promising next step to collect natural human-generated adversarial examples (Wallace et al., 2018). In the future, studying white-box (Ebrahimi et al., 2017; Bělohávek, 2017) and character level attacks (Gröndahl et al., 2018) are important directions to provide additional robustness in this critical domain.

Trained attention was explored in event detection (Liu et al., 2017), machine translation (Mi et al., 2016) and sentiment analysis (Zhong et al., 2019). More sophisticated approaches include deriving attention from human rationales (Yujia Bao, 2018). Finding better ways to make use of this kind of information is a promising avenue, especially in this low-resource domain.

Ensuring machine learning fairness has increasingly caught researchers’ attention (Corbett-Davies and Goel, 2018) since algorithms can create concerning negative societal impacts (Feller et al., 2016). Biases might be based on gender (Bolukbasi et al., 2016; Zhao et al., 2017), race (Kiritchenko and Mohammad, 2018; Rudinger et al., 2017), writing styles (Madnani et al., 2017), or other factors. More broadly, bias can also occur when annotators label the data. For example, Golbeck et al. (2017) found it hard to distinguish between mere offensive speech and hate speech.

9 Discussion

9.1 Limitations of the Annotation Instruction

The instructions for annotating rationales in tweets do not fully recognize the complex nature of language. Current annotation reduces the experts’

underlying reason to annotate a tweet as aggression to a set of key words, which is not flexible enough to convey more complicated inferences. The experts’ analyses are far more complex: they use context to disambiguate the meaning of the rationale words rather than identifying key words. Since the language is hyper-local and many of the non-aggressive tweets might appear aggressive in outsiders’ eyes, interpreting the context is a crucial component while determining an aggression label and it is important to develop more flexible context rationale representations in the future.

Take, for example, the post “I talk to alotta Da Guyz In jail Everyday On l’a Dese N***as Ain’t Sending Bro Nem A Dime 🤔 But Steady Yellin Free Em Goof ass”. At face value, the words “🤔”, “goof”, and “ass” per se do not capture the reason why this post is aggressive; the aggression label for this post in fact derives from the poster calling out the dishonesty of those who support incarcerated individuals in principle, without supporting them monetarily.

9.2 Controversial Labels

In the annotation procedure, it can be challenging to choose a single label when users express both anger and grief in a single post. In these instances, we aim to capture and annotate the tweet’s dominant theme.

For example, the post “Fuck Da System Fuck Da State Fuck 12z 🖕🤔” contains elements of aggression—the word “fuck” and the middle finger emoji—and loss—the sad frown emoji and the post’s larger context. This post fits the *aggression* label, by challenging authority and insulting, and the *loss* label, by evoking incarceration. However, a *loss* label is more suitable, because: 1) though the anger seems to target specific players, the text as a whole reveals that the poster is highlighting

institutional violence; and 2) the sad frown emoji, paired with the context that one of the poster’s loved ones was recently arrested on a drug charge (“12z” refers to law enforcement’s drug unit), cements that the user is grieving in this post.

10 Conclusions and Future Directions

In this paper, we interpret the previous state-of-the-art model, find it is biased against stop words, create a dataset of rationales, develop metrics to measure the difference between how models and humans make predictions, design an adversarial approach to test the robustness of the model, and propose a neural network architecture to incorporate rationales. Our work has an important message to the broader NLP research community that develops tools for high stakes domains: the dominant evaluation methods in current research approaches, F-score and qualitative error analysis, are far from enough to develop good systems that might potentially lead to bad social consequences. We need various different approaches to evaluate and build a model, and each of them requires further research for improvement:

Interpretability: A solid understanding of model behavior can inform researchers of the potential weaknesses and guide the process of developing corresponding solutions. When trying to interpret Chang et al. (2018)’s deep learning model, we find it biased against many function words. Still, interpretation of neural network is an ongoing research topic; even though we verified “second order consistency”, future work is needed for better and more faithful explanations.

Robustness under Adversary: It is better to break the system before it is broken when it is deployed. We find the previous model to be susceptible to insertions of certain function words, while our model is more robust against them. Nevertheless, the model is not winning every column in table 5, so there is still room for improvement.

Metrics Evaluating How Predictions are Made: We need to quantitatively evaluate the difference between how models and humans classify tweets. We develop the metrics “rationale rank” and “prediction flips under adversary”, and verify that our model behaves more similar to domain experts. F-score does not reflect this important improvement. Additionally, uni-gram (lexical level) influence is only a shallow approximation of human reasoning - a process that involves understanding the seman-

tics of a tweet as a whole and analyzing its pragmatics in the real-world context. Ideally we hope to find a formal language to approximate such a process and compare it against the model’s interpretation.

Models that Incorporate Rationales: Our new model, though simple and straightforward, successfully makes use of the rationales and outperforms the previous model under most robustness metrics proposed in this paper; we believe that more sophisticated modeling of rationales can push the performances further, both in terms of accuracy and robustness.

Feature Evaluation: Performance improvement caused by additional features does not entail that they are used by the model in an expected way. We verify the claim of Chang et al. (2018) that the pairwise interaction feature does capture user retweet/mention interactions.

Domain Experts: In our domain, stakes are high and user behavior is frequently subject to biases and misinterpretations. Therefore, we ask domain experts rather than crowd workers to extend the annotation, and ask them to provide further insights about a domain that computer scientists are usually not at all familiar with. Furthermore, we should keep a critical eye on the labels and be aware of the limitations of our evaluation metrics for model accuracy. It is convenient for computer scientists to regard the labels as ground truth, the dataset as static and try different models to improve the f-score. However, as domain experts gained a deeper understanding about the twitter users, they marked 12% of the aggression labels as controversial (section 3); the annotation may evolve over time. Indeed, we need a consistent method and dataset to compare different models fairly, but we should also keep in mind that f-score is a proxy and might not necessarily reflect an improvement in the real-world.

References

- Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.
- Petr Bělohlávek. 2017. Using adversarial examples in natural language processing.
- Terra Blevins, Robert Kwiatkowski, Jamie Macbeth, Kathleen McKeown, Desmond Patton, and Owen Rambow. 2016. Automatically processing tweets

- from gang-involved youth: towards detecting loss and aggression. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2196–2206.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.
- Serina Chang, Ruiqi Zhong, Ethan Adams, Fei-Tzin Lee, Siddharth Varia, Desmond Patton, William R. Frey, Chris Kedzie, and Kathy McKeown. 2018. Detecting gang-involved escalation on social media using context. In *EMNLP*.
- Sam Corbett-Davies and Sharad Goel. 2018. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*.
- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.
- Avi Feller, Emma Pierson, Sam Corbett-Davies, and Sharad Goel. 2016. A computer program used for bail and sentencing decisions was labeled biased against blacks. its actually not that clear. *The Washington Post*.
- Shi Feng, Eric Wallace, II Grissom, Mohit Iyyer, Pedro Rodriguez, Jordan Boyd-Graber, et al. 2018. Pathologies of neural models make interpretations difficult. *arXiv preprint arXiv:1804.07781*.
- Jennifer Golbeck, Zahra Ashktorab, Rashad O Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, et al. 2017. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 229–233. ACM.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N Asokan. 2018. All you need is” love”: Evading hate-speech detection. *arXiv preprint arXiv:1808.09115*.
- Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.
- Max Kapustin, Jens Ludwig, Marc Punkay, Kimberly Smith, Lauren Speigel, and David Welgus. 2016. [Gun violence in chicago, 2016](#).
- Svetlana Kiritchenko and Saif M Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. *arXiv preprint arXiv:1805.04508*.
- Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org.
- Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.
- Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1789–1798.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. [Learning word vectors for sentiment analysis](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.
- Nitin Madnani, Anastassia Loukina, Alina von Davier, Jill Burstein, and Aoife Cahill. 2017. Building better open-source tools to support fairness in automated scoring. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 41–52.
- Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Desmond U Patton, Sadiq Patel, Jun Sung Hong, Megan L Ranney, Marie Crandall, and Lyle Dungy. 2017. Tweets, gangs, and guns: a snapshot of gang communications in detroit. *Violence and victims*, 32(5):919–934.
- Desmond Upton Patton, Robert D Eschmann, and Dirk A Butler. 2013. Internet banging: New trends in social media, gang violence, masculinity and hip hop. *Computers in Human Behavior*, 29(5):A54–A59.
- Desmond Upton Patton, Robert D Eschmann, Caitlin Elsaesser, and Eddie Bocanegra. 2016. Sticks, stones and facebook accounts: What violence outreach workers know about social media and urban-based gang violence in chicago. *Computers in human behavior*, 65:591–600.

- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.
- Rachel Rudinger, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79.
- Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M Rush. 2019. Seq2seq-v is: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics*, 25(1):353–363.
- Eric Wallace, Pedro Rodriguez, Shi Feng, and Jordan Boyd-Graber. 2018. Trick me if you can: Adversarial writing of trivia challenge questions. *arXiv preprint arXiv:1809.02701*.
- Mo Yu Regina Barzilay Yujia Bao, Shiyu Chang. 2018. Deriving machine attention from human rationales. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*.
- Ruiqi Zhong, Steven Shao, and Kathleen McKeown. 2019. Fine-grained sentiment analysis with faithful attention. *arXiv preprint arXiv:1908.06870*.

A Supplementary Materials

A.1 Training Language Model

We first describe the language model used to generate the adversarial dataset. The language model is trained by the standard perplexity objective function, i.e., for a tweet $T = [e_1, e_2 \dots e_l]$, we want the model to predict the next token e_i based on the previous tokens $[e_1, e_2 \dots e_{i-1}]$. We train a language model for the forward and backward direction, respectively.

As for the model architecture, we embed each token to create a real vector representation $[w_1, w_2 \dots w_l]$; then we stack three layers of single direction LSTM with hidden dimension 128 on top of the embedded token representation. Then the hidden representation from the last LSTM layer is fed the final classification layer with a softmax activation, the output of which is the probability of each token in the entire vocabulary.

We randomly sampled 80% of the tweets from the unlabeled corpus (~ 1 million tweets) used in Chang et al. (2018) as our training data, and use the rest as validation data. We apply early stopping on our training by calculating the validation loss on the validation set.

A.2 Results for Other Unigrams

We ran a simple ℓ_1 logistics linear regression with unigram features on the aggression label to obtain a list of unigrams that are highly correlated with the *aggression* label. This results in ~ 300 unigrams with positive weights, and we manually selected a subset of them based on two criteria: 1) possible to insert them into a significant proportion of tweets 2) the experts do not use them to determine the label. Here we list the unigrams and their corresponding weight, and use bold front for selected unigrams. We report the adversarial flip results for other unigrams we have selected in table 12.





Unigram	Corresponding Weight
#lldv	5.755
jj	5.283
disrespect	5.218
nobody's	5.060
irritated	4.968
keta	4.864
delete	4.791
boot	4.700
snitchk	4.698
saturday	4.615
glock	4.560
opps	4.535
buses	4.427
snitching	4.418
sticks	4.396
cpdk	4.394
	4.118
,bitch	4.110
street	4.057
situation	3.881
guns	3.675
snitchkkkkkkkk	3.662
ls	3.659
rat	3.581
nuskigang	3.578
haters	3.517
kill	3.470
600	3.362
	3.362
#playcrazygang	3.324
thinkin	3.313
	3.290
police	3.284
mtg	3.065
piss	3.032
dumbass	3.023
gatta	2.972
pimp	2.942
gunn	2.843
	2.773
mitch	2.772
bust	2.748
poled	2.690
dicks	2.662
dot	2.660
cype	2.589
rayband	2.463

Table 6: Unigrams and their corresponding weights sorted in decreasing weight order

Unigram	Corresponding Weight
.m	2.448
boy	2.364
blowing	2.340
extorted	2.292
gutta	2.260
wacked	2.244
guys	2.238
wen	2.230
lame	2.164
blackin	2.145
mfks	2.138
👊	2.134
killas	2.080
😡	2.069
dead	2.043
🚫	2.029
fuck	2.022
30	1.963
n***as	1.952
👊	1.919
bih	1.910
land	1.900
thoe	1.875
dthang	1.852
800	1.851
bout	1.810
strapped	1.807
repping	1.786
turn	1.767
shoutout	1.693
lucky	1.680
basically	1.674
ill	1.669
mfs	1.667
😓	1.632
shut	1.631
crease	1.615
page	1.603
shirt	1.583
punching	1.561
boa	1.549
niqqas	1.529
fight	1.517
👉	1.466
n***az	1.463
riot	1.453
ebt	1.444

Table 7: Unigrams and their corresponding weights sorted in decreasing weight order (continued)

Unigram	Corresponding Weight
fee	1.438
blk	1.437
pole	1.405
o	1.377
👹	1.374
...lol	1.369
hoes	1.362
doer	1.348
pussy	1.347
wanna	1.337
gotta	1.331
another	1.315
let	1.303
mean	1.297
watchem	1.274
odee	1.202
ready	1.196
ass	1.191
into	1.175
@	1.113
put	1.106
learn	1.105
line	1.100
yea	1.087
👊	1.074
kids	1.040
P	1.037
tell	1.001
frontstreetsavagesquad	0.981
got	0.970
fucked	0.962
run	0.959
screaming	0.944
changed	0.940
delusional	0.940
#tyquanworld	0.910
any	0.909
talking	0.878
wassup	0.866
💣	0.854
on	0.846
😓	0.845
thats	0.844
name	0.826
life	0.811
hate	0.810
lazy	0.803

Table 8: Unigrams and their corresponding weights sorted in decreasing weight order (continued)

Unigram	Corresponding Weight
much	0.799
mino	0.791
!url	0.788
!	0.785
knew	0.747
tire	0.698
bro	0.691
dnt	0.688
😏	0.723
ae_	0.687
🔴	0.678
\$	0.658
wid	0.657
pictures	0.654
game	0.651
ever	0.642
st	0.635
n***a	0.625
world	0.616
went	0.616
who	0.614
tw	0.613
😭	0.601
anotha	0.578
a	0.571
👉	0.565
frm	0.563
die	0.562
mufuka	0.535
irrelevant	0.525
👊	0.523
yall	0.512
bet	0.511
out	0.508
snoop	0.491
betta	0.491
girl	0.480
@user	0.478
that	0.460
😬	0.458
ain't	0.447
watch	0.434
torrance	0.424
beef	0.415
mouth	0.407
mom	0.406
🐱	0.400

Table 9: Unigrams and their corresponding weights sorted in decreasing weight order (continued)

Unigram	Corresponding Weight
o'hare	0.394
he	0.391
's	0.388
yo	0.387
😏	0.385
forever	0.381
#nolacking	0.375
🐱	0.373
snitched	0.360
gave	0.355
if	0.354
talk	0.352
keep	0.329
as	0.327
internet	0.325
da	0.323
producer	0.323
gang	0.322
opp	0.314
chica	0.301
homie	0.297
onna	0.289
beat	0.288
bitch	0.287
😂	0.283
casper	0.283
#loony	0.269
hoe	0.260
whoever	0.255
shootin	0.252
#mob	0.197
lady	0.183
rose	0.183
fake	0.181
we	0.178
gun	0.177
be	0.176
smoked	0.175
#8tre	0.170
caught	0.162
jarocity	0.155
!	0.153
😏	0.137
shordty	0.126
just	0.104
everyday	0.090
nail	0.076

Table 10: Unigrams and their corresponding weights sorted in decreasing weight order (continued)

Unigram	Corresponding Weight
snitch	0.073
bumming	0.071
of	0.070
ona	0.067
u	0.058
steady	0.052
like	0.050
talm	0.048
ways	0.047
#wuggaworld	0.044
soft	0.042
#rt	0.041
yellin	0.041
do	0.037
in	0.035
#cmb	0.034
#cvg	0.029
👎	0.027
no	0.025
#051ym	0.025
tutugang	0.018
#6775	0.010
🧠	0.008
#jar	0.005
goof	0.005
dime	0.003
#bricksquad	0.003
📺	0.001

Table 11: Unigrams and their corresponding weights sorted in decreasing weight order (continued)

Models \ Unigrams	in	be	do	any	u	out
Blevins et al. (2016)	3.68	0.0*	0.0*	39.92 [!]	48.08 [!]	25.64 [!]
Chang et al. (2018)	7.56 [!]	6.56 [!]	0.4	9.2	6.84*	3.6
CNN + Twitter	1.88*	1.48	0.84	1.16*	13.88	1.8
LSTM	4.24	2.72	1.08 [!]	8.36	16.12	2.48
LSTM + Rationale	3.96	1.12	0.08	3.64	12.76	1.4*

Models \ Unigrams	basically	yea	ever	UPI	👉	we
Blevins et al. (2016)	206.08 [!]	67.84 [!]	47.64 [!]	106.36 [!]	49.84 [!]	69.6 [!]
Chang et al. (2018)	3.64*	2.96	2.04	3.56	2.28*	16.04
CNN + Twitter	5.4	1.52*	1.04	3.08	3.16	10.52*
LSTM	7.44	13.6	2.28	1.64	9.68	16.92
LSTM + Rationale	4.16	8.0	0.16*	0.48*	3.92	11.96

Table 12: The number of model’s prediction flip from non-aggressive to aggressive, out of the 800 attacking tweets generated by inserting a specified “neutral” unigram. To obtain a stable estimate, the count of number of flips of each model is averaged across 5 model runs and models trained on 5 folds - thus leading to non-integer results. For each column, the worst performing entry is marked with “!” and best with “*”.