Eliminating Bias in AI: Compensating for Label Bias in Data Sets through Re-Annotation

Diego Jurado
University of Pittsburgh
Department of Computer Science
diego.jurado@pitt.edu

Abstract

The data sets that are most widely used for automatic hate speech detection contain Tweets or other social media-based messages, that are manually judged and annotated by large groups of people. These annotators are human, they are prone to bias, we investigate the task of modifying manually-annotated data with a given rule set, and observe how it influences a classification task using Support Vector Machines (SVM). We first re-annotate randomly, then we introduce an intelligent method, and compare the performance of these two implementations to a baseline binary classification SVM, arguing that the closer a given incorrect prediction is to the decision boundary of an SVM, the more that Tweet or message was originally incorrectly annotated.

1 Introduction

With the increasing importance of "big data" in our society it becomes imperative to perform ethical analyses on our systems and urge necessary change. When implementing downstream tasks like classification, where the priority is currently found on accuracy, precision, recall or F1 Score, one will typically search for a large and reliable data set for the specific task. However, there has yet to be an established standard for data set annotation, much less tools meant to achieve a standard.

While humans can essentially switch between different language types like regional dialects, a computer is not necessarily able to. So, to maintain a high level of accuracy for a given task for example summarization for a news source, it would not be in the developer's best interest to use a Twitter-based data set. This choice leads to a phenomenon referred to as selection bias (Søgaard et al., 2014). While the issue of selection bias can be avoided by simply finding a more apt to task data set or model, the issue with label bias is a bit more difficult to manage. While the current trend is to be more prudent in annotation projects, this solution falls short for the developers who lack resources to annotate new data (Snow et al., 2008). In addition to increased prudence, a tool kit should be created to assist to incorrectly annotated data compensation. In big tech especially these data sets are being used for commercial applications, which may or may not be ethical, however, the focus is on the impact the data may have itself. Incorrectly labeled data will lead to incorrect classification, and consequently less than acceptable results.

The current trends in annotation projects demonstrate in the initial phase there is much more prudence and care place in comparing annotators' performance, where the project managers compute agreement scores, select reliable annotators, and elaborate on annotation guidelines. While this plays a role in eliminating much of the annotators' individual biases, it has yet to reduce all of it. Using non-expert annotators, increases label bias. Alternatively, crowd-sourcing services like Amazon's Mechanical Turk can successfully be used for annotation project creators to be in control of labels, yet even then the data set becomes victim to the creators bias (Jha et al., 2010). However, on way to reduce this bias is through the collection and averaging of multiple annotations on the same data point. This method is referred to as majority voting and is analogous to using ensembles of models to obtain more robust systems (Søgaard et al., 2014). I propose a method of compensating for incorrect annotations after the completion of an annotation project. The project will observe the results of re-annotating data using different methods for acquiring percent error for calculating the portion of the data set that will be re-annotated, and sampling which data points should be relabeled.

Classifiers multiply and reflect the human biases that go into creating the data set. We know that the definition of hate speech is rather subjective, however, there is common ground between these definitions. This common ground can be seen as

an objective standard for classifying hate speech. I propose the following two conclusions: If the data is correctly annotated to the objective standard, then any change in annotation would reflect negatively on the classifiers end, because the patterns being observed by the classifier would shift to being more subjective and inherently not be as accurate as a result. Additionally, if the data is incorrectly annotated, then any change in annotation would reflect positively on the classifiers end, because the patterns being observed by the classifier would shift to being less subjective and inherently being more accurate as a result.

2 Related Work

The problem of detecting hate speech on social media has had maintained interest from the NLP and AI community. Most of the systems currently being utilized require immense data sets and ridiculous training times to keep up with current benchmarks (Strubell et al., 2019). While most approaches to automated-hate-speech-detection take into account only features related to the single message to be classified, recent studies have highlighted the importance of context in interpreting the meaning of a message and its possible hateful content (Lee et al., 2018; Kshirsagar et al., 2018; Zhang et al., 2018; Park and Fung, 2017). Moreover, there has been work done regarding manual re-annotation of data to see its influence on contextual deductions, where they conclude context is necessary to understand the real intent of the user, and that basing the classification solely off the content might return an incorrect classification (Menini et al., 2021). This group has essentially proved that re-annotating plays a role in revealing the true meaning of a tweet, when context is not available.

There has also been work done in looking at different data set biases throughout the development process, while on of the most important, selection bias is a problem, this is easily solved with experience in the field and does not plague experienced researchers or developers. However, a more pressing issue that does not yet have an ideal solution is the task of label bias, in other words, the bias each annotator incorporates into the data set through annotation whether intentional or not (Søgaard et al., 2014).

3 Methodology

3.1 Classification Algorithm

A Binary SVM is a classifier which discriminate data points of two categories. Each data object (or data point) is represented by a n-dimensional vector. Each of these data points belong to only one of two classes. A linear classifier separates them with a hyperplane. SVMs achieve high generalization by maximizing the margin, as well as support an efficient learning of nonlinear functions using the kernel trick.

In the experiment a Binary SVM is being utilized to detect whether a given input tweet is considered hate speech or not. Further, we use SVM as a baseline to compare the performance of the modified data sets. Moreover, SVMs are accurate, as well as relatively computationally efficient, leading to shorter training times, and reduce carbon output, as opposed to training a similar quantity of deep neural models.

3.2 Randomized Re-Labelling

The Randomized Re-Annotation Scheme is a method being used to test the volatility of SVMs and the corresponding data set being used. Essentially, the re-annotation scheme samples random tweets and re-labels them as the opposite class. Considering it is randomly re-labelling tweets, there is a chance it manages to re-label optimally, however, it will often re-label in a sub-optimal manner. Assuming an optimal re-label, during training the hyperplane would fit optimally, and in the testing would classify data set optimally, however, this blurs the line between over fitting and not, as over fitting typically refers to the model and not the data itself.

Most of the tweets classified by the SVM that fall into the set of false negative, and false positive categories are typically farther from the decision boundary than the true positive and true negative. So, if it randomly samples farther from the decision boundary, these tweets may cause the decision boundary to lean more towards a specific prediction class, resulting in prediction bias, causing either the true negative or true positive class to suffer.

3.3 Intelligent Re-Labelling

The Intelligent Re-Annotation Scheme is a method being used to “optimally” re-label a given tweets class to attempt to compensate for the percent error of a given data sets annotation method. Essentially,

the re-annotation scheme samples the n tweets that are closest to the decision boundary that correspond to the "hate speech" class, and re-labels them as "not-hate". Additionally, the re-annotation scheme samples m tweets that are closest to the decision boundary that correspond to the "not-hate" class, and re-labels them as "hate speech".

Considering, the re-annotation scheme samples those tweets closest to the decision boundary, one could expect during training that this would create a finer decision boundary, essentially dividing the classes more optimally, as it is creating a greater margin, and adding more support vectors. Overall, the further maximization of the separation between classes minimizes the prediction bias present in the classifier.

3.4 Quantification Methods

The method used for calculating the quantity of tweets to re-annotate is based off an initial SVM confusion matrix score, where the false negative and false positive rates are then multiplied to the corresponding section of the overall data set to find the number of tweets needed to re-annotate. Assuming the SVM is a perfect technology then it demonstrates the error in the data set through the error in the corresponding confusion matrix, namely the false positive and false negative ratios, the error in the test data set is proportional to the rest of the data set, thus demonstrating a need to re-annotate.

Quantification Formulas

From "hate" to "none":

$$q_1 = \lfloor |D_h| * r_n \rfloor$$

From "none" to "hate":

$$q_2 = \lfloor |D_n| * r_p \rfloor$$

4 Experiments

4.1 Preprocessing

Before running the classification experiments, we pre-process the tweets gathered from the Hate-Speech-Annotation data set (Waseem and Hovy, 2016) as follows: we re-label all different hate speech labels from *"racism"*, *"sexism"*, etc. to a generic class label *"hate"* in order to form a binary class data set. Then, considering the data set consists of around 11,000 tweet samples with an uneven split, we randomly sample an even amount of each class, 3,000 of each, for each trial we run. Upon having an even split of data, we then convert each sample to a vector using Scikit-Learn, a Python library built for predictive data analysis. Since SVM takes in input sentence embedding, we convert the context and the current tweet into sentence embeddings that encapsulate 5,000 maximum features per tweet.

4.2 Experimental Setup

The main assumption driving the experiment is that the tweets closest to the decision boundary following the creation of an SVM correlate highest to the support vectors, therefore the tweets closest to the decision boundary and are in either the false negative or false positive sets actually correspond to the other class label, and we deduce they were initially labeled incorrectly, while those furthest from the decision boundary are correctly placed, yet the features to classify them correctly are not being identified.

Each model configuration is evaluated randomly dividing the same 6,000 tweets into training (80%) , and test (20%) sets. Each configuration is tested 30 times each with a randomly selected 6,000 tweets, and the results are averaged.

Prior to the experiment we tested different types of kernels (i.e. linear, polynomial or radial) in order to find an optimal SVM as a baseline, the experiment reports only the results obtained with this best configuration, i.e. linear kernel and C = 1.0.

5 Evaluation

The overall trend demonstrated in Table 1 by the randomized re-annotation method is around an 8.5% decrease compared to the average performance of the baseline SVM model. Ultimately, the decrease in performance is most likely evidence of the sub-optimization of hyperplane formation during training, because the volatile nature of the randomized annotation scheme tends to have bias towards one class, essentially shifting the decision boundary and causing the support vectors to be sub-optimal regarding feature identification and subsequent classification. On the other hand, the trend demonstrated by the intelligent re-annotation method is around 3.5% increase compared to the average performance of the baseline SVM model. Essentially, the increase in performance is most likely evidence of a more optimal hyperplane for-

Model	Precision	Recall	F-Score	Precision Δ	Recall Δ	F-Score Δ
Baseline SVM	0.81701	0.81600	0.81595	—	—	—
Randomized	0.72989	0.72878	0.72851	−0.08712	−0.08722	−0.08744
Intelligent	0.85354	0.85300	0.85292	+0.03653	+0.03700	+0.03698

Table 1: Classification Performance Results on Hate-Speech-Tweet Data Set, average of 30 runs. Precision, Recall, and F-Score are weight-averaged. Delta change is taken from subtracting baseline performance from a new model.

mation during training. Through the re-annotation of data that is closest to the baseline hyperplane it is likely that these recently re-annotated tweets offer new features or more evidence to certain features that help in the identification of computational patterns, which lead to a more refined hyperplane, and overall, a better classifier.

6 Discussion

The automatic re-annotation of data deserves further study to develop better methods. More so, the current method, while successful to some degree, it may be worth considering to develop more intelligent methods that may incorporate elements of neural networks for feature analysis and re-prioritization.

6.1 Applications

Note that the method should not be used as a general optimization of an SVM or any other classifier, as that is not what it is. This is a method of data annotation correction, meaning it is meant to target the percent error within the data set development process. Each data set can only be "optimized" or "corrected" once. If the method is used multiple times on the same data set it can be inferred that this process would multiply and reflect the original bias rendering the data set unusable, as it would resemble a typical classifier over-fitting to its data set.

6.2 Ethical Considerations

6.2.1 Environmental Concerns

Note one of the reasons influencing the decision to experiment with SVM as opposed to any other form of classifier is that they are relatively computationally efficient, high accuracy, and subsequently "environmentally-friendly". Emma Strubell and their team state that a concentrated effort should be placed in researching and finding hardware and software that are computationally efficient, specifically AI models (Strubell et al., 2019). However, a different interpretation of this statement could be to find computationally efficient optimizations in any aspect of AI, and we look into data "optimization" through the correction of annotations. So, if there is a method that can be used to reduce the number of annotators needed, or reduce the amount data needed to verify classification features, it can be regarded as a relatively "environmentally-friendly" method, it may still requires a similar amount of computational power to run once, however, once is all it needs. Ultimately, striving for a more intelligent data set annotation corrector will be a positive future research direction.

6.2.2 Downstream Impact

Currently the method proves successful to a certain degree. However, use of the current method could lead to temperamental results in different configurations. While it may increase performance of an SVM on this data set, that may not be the case for every data set or classifier algorithm. The underlying intention of the method is to reduce bias in the data set, thus increasing general performance on a classifier, yet it may produce the exact opposite response in different data sets. Hence, this method should not be used in any commercial sense until further testing or development is performed.

7 Conclusion

Researchers should prioritize data set optimization.

Our experiments demonstrate that it would be beneficial to directly compare different combinations of data sets and classifiers. Moreover, it can prove beneficial to create an open-source tool kit that can be used to raise the quality of a data set without the need of human labor. This does not refer to the task of classification itself as that would be counter intuitive. We still require human labor to perform an initial annotation, however, we can then computationally optimize the data set using more intelligent re-annotation methods.

Data set developers should be transparent with their process.

While the re-annotation method targets the percent error in a given data set, and aims to re-annotate in order to correct this error, it would be significantly more easy to develop a more specialized method if we could calculate the exact percent error of a given data set. However, one could only accomplish this if we have complete transparency to everything that happened on a given data set.

References

Mukund Jha, Jacob Andreas, Kapil Thadani, Sara Rosenthal, and Kathleen McKeown. 2010. Corpus creation for new genres: A crowdsourced approach to pp attachment. pages 13–20.

Rohan Kshirsagar, Tyrus Cukuvac, Kathy McKeown, and Susan McGregor. 2018. Predictive embeddings for hate speech detection on Twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 26–32, Brussels, Belgium. Association for Computational Linguistics.

Younghun Lee, Seunghyun Yoon, and Kyomin Jung. 2018. Comparative studies of detecting abusive language on twitter. *CoRR*, abs/1808.10245.

Stefano Menini, Alessio Palmero Aprosio, and Sara Tonelli. 2021. Abuse is contextual, what about nlp? the role of context in abusive language annotation and detection. *CoRR*, abs/2103.14916.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on Twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45, Vancouver, BC, Canada. Association for Computational Linguistics.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii. Association for Computational Linguistics.

Anders Søgaard, Barbara Plank, and Dirk Hovy. 2014. Selection bias, label bias, and bias in ground truth. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Tutorial Abstracts*, pages 11–13, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. *CoRR*, abs/1906.02243.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Ziqi Zhang, David Robinson, and Jonathan A. Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *ESWC*.