2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-6654-6283-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMLA55696.2022.00252

Semi-Supervised Machine Learning for Analyzing COVID-19 Related Twitter Data for Asian Hate Speech

Caitlin Richardson
Department of Mathematics
University of Tampa
Tampa, Florida
Caitlin.richardson@spartans.ut.edu

Sandeep Shah
Department of Computer Science
North Caroline Agricultural and Technical State University
Greensboro, North Carolina
sshah@aggies.ncat.edu

Xiaohong Yuan
Department of Computer Science
North Carolina Agricultural and Technical State University
Greensboro, North Carolina
xhyuan@ncat.edu

***Abstract*— The COVID-19 pandemic left a lot of people sick, tired, and frustrated. Many people expressed their feelings on social media through comments and posts. Detecting hate speech on social media is important to help reduce the spread of racist comments. Machine learning algorithms can be used to classify hate speech. In our experiments, we implement semi-supervised machine learning algorithms to classify Twitter data. We used a count vectorizer as the feature and a support vector machine (SVM) classifier to classify COVID-19 related Twitter data while changing the amount of labeled data available. We found that self-training semi-supervised machine learning has similar effectiveness to supervised learning when there is significantly less training data available.**

Keywords*— *Asian hate, machine learning, self-training, semi-supervised, Twitter

I. Introduction

Technology has become increasingly relevant to our everyday lives especially after the COVID-19 pandemic hit. Social media is one of the ways that others remain connected to each other at a time when no one could meet up in person. As beneficial as social media is to stay connected, it also allowed the public to express their view on the entire situation. In some cases, people would make very aggressive posts, racist comments, and spread hate speech. More specifically, Asian hate speech began to rise as some individuals began to blame coronavirus on Asians during the pandemic [1, 2].

Being able to manually remove or report all hate speech on social media is unrealistic and costly. In this paper, we apply machine learning algorithms to classify tweets as hate, counter hate or neutral speech. The method of using manually labeled data to train machine learning models is known as supervised learning [3]. Unsupervised learning is when there are no labeled data to train a model on [4]. In this paper, we focus on semi-supervised learning which is between supervised and unsupervised learning. The idea of semi-supervised learning is that you can train a model on a small portion of labeled data, and the model is then used to predict the outcome on a large set of unlabeled data. Since it is costly to have experts manually label large amounts of data for training, it is more practical to label a small amount of data for training and use semi-supervised machine learning methods.

There are two main methods for conducting semi-supervised machine learning. There is self-training and co-training. Self-training is a simple and effective method of semi-supervised machine learning [5]. The method revolves around training a base classifier on a small amount of labeled data. Then apply the model to unlabeled data and predict the probabilities. From there, only add the labeled data with high probabilities to the training data while also removing it from the unlabeled group. Retrain the classifier and repeat until there are no more high probability predictions left. Lastly, evaluate the classifier on a test dataset.

Co-training was proposed by Blum and Mitchell [6]. In co-training, they assume that the data can be broken into two sets, each new set is substantial enough to train a classifier, and the two sets are independent of each other. The goal for co-training is that you can train two separate classifiers on your two subsets and then be able to use the predictive models on the unlabeled data set and each classifier will add the unlabeled data to the new training set according to its view.

Support Vector Machine (SVM) is one of many machine learning classifiers and is the chosen classifier for our research. Essentially, SVM creates a hyper-plane to separate the data into classes. In two dimensions, the hyper-plane is a line. The line could be linear or non-linear [7]. Support vectors will be created which are points that are closest to the hyper-plane. The goal is for the hyper-plane to be about the same distance from each support vector of each class, which is also known as maximizing the margin. SVM also protects itself against overfitting [8] so you could deal with a large feature selection since all the data is not required to fit within the function.

In this paper, we explore how semi-supervised machine learning can be implemented in analyzing hate speech from social media. With semi-supervised machine learning, we would be able to use a small portion of labeled data to predict a large portion of unlabeled data. Our main goal is to determine how effective self-training based semi-supervised learning is compared to supervised learning, and how the amount of training data used affects the performance of the semi-supervised learning. For the data in this experiment, we use tweets from Twitter during the height of the COVID-19 pandemic to train our model to classify each tweet as Hate, Neutral, or Counter hate. More specifically our dataset will focus on Asian hate during the pandemic [1].

The rest of the paper is organized as follows: Section II presents related research to supervised and semi-supervised machine learning in tweeter data analysis and hate speech detection. Section III explains the experiments we conducted to classify COVID-19 related tweets. Section IV presents the results of each trial and the comparison among all trials. Section 5 draws conclusions and discusses future work.

978-1-6654-6283-9/22/$31.00 ©2022 IEEE
DOI 10.1109/ICMLA55696.2022.00252

II. Related Work

This section reviews related work on using supervised and semi-supervised machine learning for analyzing tweets and hate speech detection.

A. Supervised Machine Learning

Davidson et. al [9] used supervised machine learning to classify twitter data as hate, offensive, and neither. His data was randomly selected 25k tweets from a larger Twitter API database consisting of 85.4 million tweets. From there they had multiple people annotate the tweets and kept the label of the tweets if many of the annotators agreed on the label. From there, their data was split into 90% train and 10% test. They assessed five different commonly used models which include logistic regression, naive Bayes, decision trees, random forest, and linear SVMs with TF-IDF as the vectorizer. They found logistic regression and SVM performed significantly higher than the rest but chose logistic regression to use as their final classifier. The results for correctly predicting hate speech were 61%, offensive speech at 91%, and neither at 95%. Their model had difficulty differentiating between hate speech and offensive speech. Most often, if the tweet had any racial or homophobic slur the model would automatically label it as hate. They found that racist or homophobic slurs were considered hate speech but being sexist was just considered offensive. Their model was good for classifying racial hate but held bias towards areas like sexism.

Shah et. al [8] also used supervised learning to classify hate-related tweets. He experimented with multiple classifiers to compare the accuracy and F1 scores of classifying the tweets. The classifier that produced the best result was the Support Vector Machine classifier with a 70% accuracy. He also compared two different vectorizers to see which one resulted in higher accuracy. He used a count vectorizer which counted how many times each word appeared in the tweet. He also used a TF-IDF vectorizer which finds the frequency of a word and returns a score of each word present in the dataset. Overall, the count vectorizer performed slightly better than the TF-IDF vector by a few percent in each classifier assessed. The dataset was collected by Ziems, Caleb, et al. [1] where there were over 206 million tweets from January 2020 to March 2021 among which 2,291 were annotated. We built upon this dataset to add more annotated data to produce our dataset.

B. Semi-supervised Machine Learning

Alsafaria and Sadaouri [10] used semi-supervised learning to predict among 5 million Arabic tweets for hate speech. More specifically they implemented self-training models that iteratively repeated the process to improve results for classifying hate speech. After each iteration was finished, any pseudo labels with a high probability of 0.999 were added to create a new training dataset to retrain the classifier. They compared four different models to see which produced the best results. There were two Deep Neural Networks (DNNs) and two different text vectorization algorithms (TVAs) and they experimented with the four combinations. The DNNs used were a Convolutional Neural Network (CNN) and a Bidirectional Long Short-Term Memory (BiLSTM) matched with both AraBert (Bert) and Skip-gram W2V (SG) embedding. Their data had 9,345 labeled data, five million unlabeled data, and 4,002 testing data samples. Out of the four models they found that CNN with SG yields the best outcome because of the efficiency in inference speed and size. It also had the highest scores across all performance metrics with an F1 score of 88.59. They only ran their trial for five iterations because of how long it takes for one iteration to occur. When stopped at the fifth iteration their training corpus has grown from the original 9,338 tweets to 17,646 tweets. If they wanted to continue the process to enlarge their corpus more they could, but were satisfied with their results. Their experiment showed the promising results of using a CNN with SG as the most effective DNN and TVA paired together.

Gao et. al [11] were able to implement self-training and an improved self-training model when analyzing tweets from Twitter for their credibility. He experimented with 10%, 20%, and 30% labeled data. The improved self-training model used a repeated labeling strategy to help with accuracy and possible mistakes from pseudo-labels. He used logistic regression as his base classifier and then verified his results with a random forest classifier. Thirty percent of training data on his improved model was shown to create the best results. With the random forest model, the f1-score improved from 0.84 to 0.9 and the confusion matrix showed that 98% of the data was correctly labeled as not credible and 91% was correctly labeled as credible. The improved self-training with a repeated labeling strategy helped reduce the amount of data left untouched while also improving the overall results.

III. Methodology

This section provides an overview of how the dataset was prepared, processed, and evaluated for hate speech classification of Twitter data.

A. Dataset

Our dataset was initially created by Ziems, Caleb, et al [1] where it contained 206 million tweets. The data was collected from January 2020 to March 2021. 2,291 tweets within the dataset were manually labeled as hate, counter hate, and neutral. From there we expanded the dataset by adding 2,220 pieces of labeled data. We defined hate as any threatening or ill-intended text aimed at Asians. Neutral was considered as any text that held no threat or harassment and counter hate was considered any speech that protected and defended Asians against any hatred. Two students manually labeled 1,110 pieces of data each that were selected from the larger dataset. Then we added to the previously existing labeled dataset creating a total of 4,511 tweets that were used in the following semi-supervised machine learning algorithm. Fig. 1 shows the distribution of tweets with their labels that make up the entire dataset we worked with.

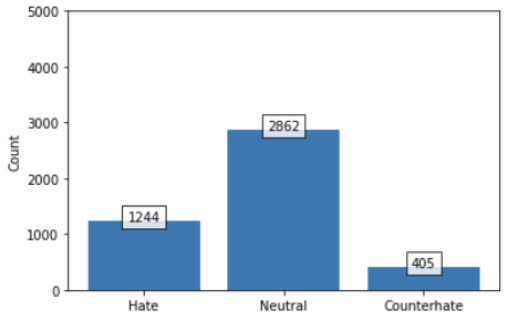


Fig. 1. The distribution of tweets with their labels in the Dataset

B. Data Pre-processing

The data were loaded into a data frame where the first column represents the tweet and the second column represents the label of either "Hate", "Neutral", or "Counter hate". We begin by first changing the label column into 0, 1, or 2 to align respectively with the labels "Hate", "Neutral" and "Counter hate". From there we must remove any stop words in the text before vectorizing because stop words have no effect on if the tweet is considered "Hate", Neutral", or "Counter hate". Stop words can be "are", "is", "or" etc. They give no indication of value to how the tweet can be classified therefore we remove them before processing. Then we transform the tweet column from text data to numerical values to train our model. We use the count vectorizer to transform the data because Shah et al. [8] showed that it performed better than TF-IDF vectorizer for the dataset. The count vectorizer creates a vector based on the frequency of each word showing up in the given text. Once all text data in the data set is transformed into numerical values, we randomized the dataset. This was done to ensure there were no groupings or distinct patterns when the data was originally inputted. Table I shows how a tweet is pre-processed through an example.

TABLE I. DATA PRE-PROCESSING EXAMPLE

Tweet	miss corona here killed thousands of people worldwide and is keeping us in our shitty little homes... shes so cancelledt!
Tweet after removing stop words	Miss corona killed thousands people worldwide keeping shitty little homes shes cancelledt
Count Vectorizer	1 1 1 1 1 1 1 1 1 1 1 1

C. Self-training

Self-training is one method of semi-supervised learning. The idea of self-training is that first the dataset is split into test data, training data, and unlabeled data. Then the classifier is trained on the training data. After that, the model is used to predict the probability of the unlabeled data and the data that has a high probability of classification is selected. The unlabeled data with high probability will then be added to the training data set with their predicted label. After that, we re-train our classifier on our new training data and repeat the process until there are no high probability predictions left. The self-training process is shown in Fig. 2. The self-training algorithm is shown in Table II.

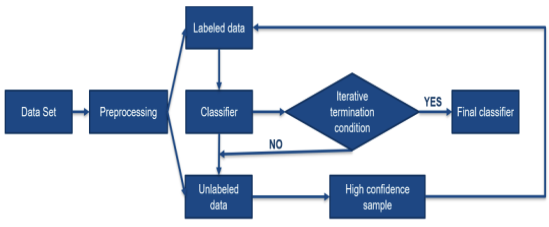


Fig. 2. Self-training Process [11]

TABLE II. SELF-TRAINING ALGORITHM

Steps
Step 1: Split data into a test, train, and unlabeled sets
Step 2: Train classifier on training data
Step 3: Use the trained model to predict on unlabeled data set, the highest probabilities are now considered "pseudo labels"
Step 4: Add "pseudo labeled" data to training data and remove them from unlabeled data
Step 5: Repeat steps 2-4 until no more "pseudo labels"
Step 6: Evaluate the model on test data

IV. RESULTS

The dataset consists of 4,511 tweets from Twitter during the pandemic. All the tweets are from the COVID-19 pandemic from January 2020 to March 2021when Asian hate speech began to rise. The tweets were annotated to be classified as Hate, Neutral, and Counter hate. We used the count vectorizer to transform the data from text to numerical values to be processed. Then we used the Support Vector Machine classifier to classify the unlabeled data based on the labeled data given. We chose to use SVM as our classifier because Shah et al. [8] showed that SVM produced the highest accuracy out of seven machine learning algorithms. For the results, we labeled "Hate", "Neutral", and "Counter hate" as "0", "1", and "2" respectively.

A. Classification Performance Metric

To measure the performance of the self-training algorithm, we use Accuracy, Precision, Recall, and F1-score.

Accuracy is the measure of total correctly labeled samples over the total amount of samples represented as a percentage.

Precision is the correctly labeled sample out of the total predicted samples classified as the same label.

Recall is the correctly labeled sample out of the actual labeled samples with the same label.

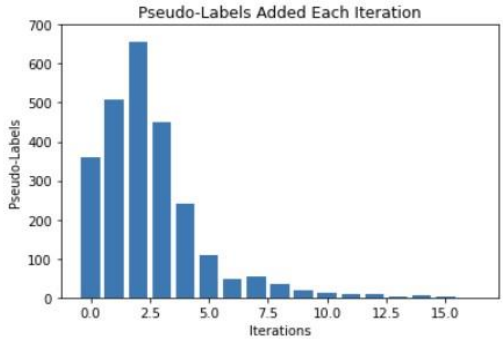
F1 score is the harmonic mean between precision and recall.

B. Self-Training Algorithm

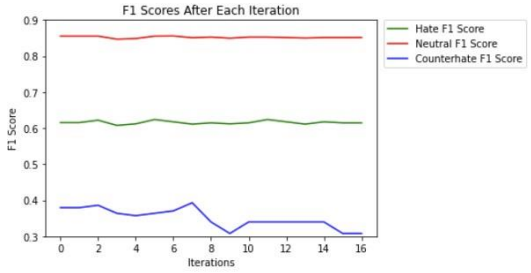
After randomizing the data, we kept our test data constant at 10% of the labeled data for each trial. We would change the amount of data used for initial training of the SVM classifier in the self-learning algorithm from 15%, to 20%, and then 25% of the labeled data respectively. Thus, the amount of unlabeled data is adjusted to the remaining percentage left. Since all our data is annotated for the unlabeled portion, we dropped the label of the tweets to act like the data is unlabeled. We also used a probability threshold value of 0.9 or 90%. So, any data that has a probability above 90% confidence in a certain predicted label, would be added to our training data and used to re-train the model while also simultaneously removing the data from the unlabeled dataset. This would repeat until there are no longer any high probability predictions left. When the loop ends the model will then be evaluated on the test data to measure the performance of the newly built model. Also, the amount of unlabeled data is produced at the end of the loop which is the length of the unlabeled data list. We used 10-fold cross-validation to validate the training model.

1) Training with 15% Labeled Data

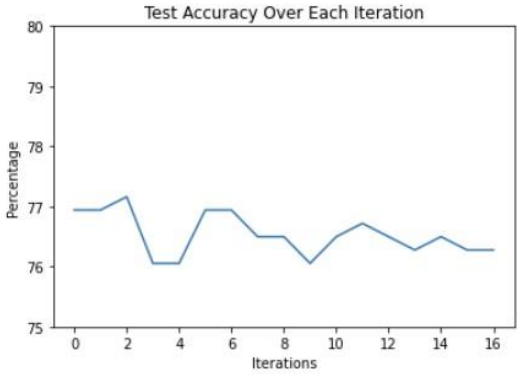
After 16 iterations of the self-training algorithm, the results are shown in Fig. 3.



(a) Pseudo labels added to the training data after each iteration



(b) The F1 scores of Hate, Neutral, and Counter hate label after each iteration when evaluated on test data



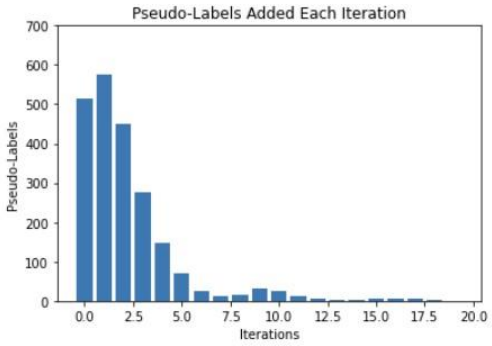
(c) The accuracy of each iteration after the training data increased and the model was re-trained

Fig. 3. Results with 15% Labeled Data for training after each iteration: (a) pseudo labels (b) F1 Scores of each label (c) Accuracy evaluated on test data of model

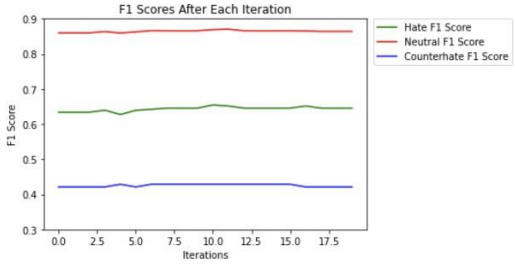
After many iterations, the pseudo labels added increased in size before gradually dropping until there are no more pseudo labels. The F1 Scores for Hate and Neutral stay the same while the Counter hate F1 score decreases after each iteration. The test accuracy is decreasing after many iterations.

2) Training with 20% Labeled Data

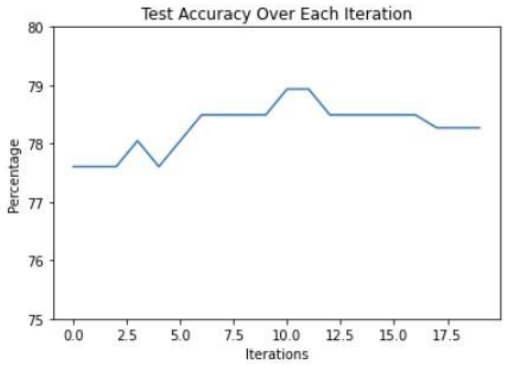
After 20 iterations of the self-training algorithm, the results are shown in the figure below.



(a) Pseudo labels added to the training data after each iteration



(b) The F1 scores of Hate, Neutral, and Counter hate label after each iteration when evaluated on test data



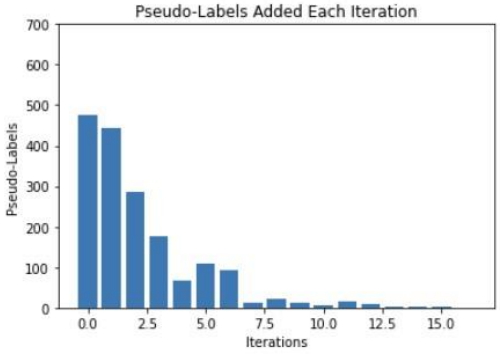
(c) The accuracy of each iteration after the training data increased and the model was re-trained

Fig. 4. Results with 20% Labeled Data for training after each iteration: (a) pseudo labels (b) F1 Scores of each label (c) Accuracy evaluated on test data of model

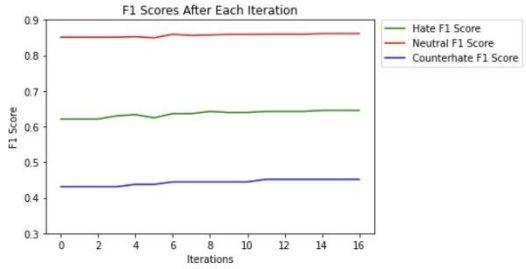
As the data increases the pseudo labels have one spike after the first iteration and then the amount of pseudo labels steadily decreases. All the F1 Scores of each label seem to either remain the same with a little fluctuation or a slight increase. The test accuracy appears to increase before decreasing and then plateauing.

3) Training with 25% Labeled Data

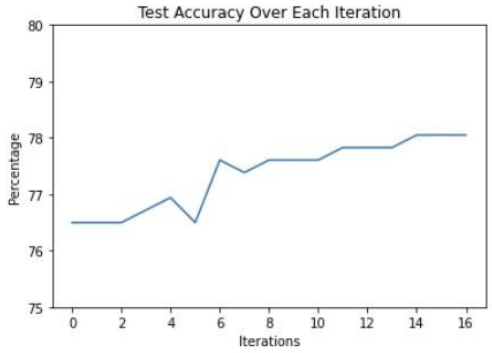
After 16 iterations of the self-training algorithm, the results are shown in the figure below.



(a) Pseudo labels added to the training data after each iteration



(b) The F1 scores of Hate, Neutral, and Counter hate label after each iteration when evaluated on test data



(c) The accuracy of each iteration after the training data increased and the model was re-trained

Fig. 5. Results with 25% Labeled Data for training after each iteration: (a) pseudo labels (b) F1 Scores of each label (c) Accuracy of the model evaluated on test data

As the iterations increase the pseudo labels decrease and the F1 scores of each label increase. The test accuracy appears to be a steady increase after multiple iterations.

C. Comparison of Results

Table 2 compares the performance of the self-training algorithm with different percentages of labeled data for initial training. We compared precision (P), recall (R), F1 score (F), test accuracy, cross-validated (CV) accuracy, and training accuracy. The top row is supervised learning as 80% were labeled data for training and 20% were test data. Supervised learning is used as a comparison to see the effectiveness of semi-supervised learning in the different amounts of labeled data. As seen in the table below, F1 scores and accuracy see the most noticeable change as the percentage of training data increases in the self-training method. The F1 scores increase for every label but more notably the F1 score of counter hate improved the most. The F1 score goes from 0.31 (with 15% training data) to 0.45 (with 25% training data) which is only 0.01 away from being the same as the supervised counter hate (2) F1 score. Besides the minuscule difference, the F1 scores of hate (0) and neutral (1) are the same in the 20% and 25% training data as the supervised model F1 scores if not slightly better. The test accuracy is also higher in the 20% and 25% training data than in the supervised model which has 80% training data.

TABLE III. RESULTS FOR EACH TRAINING DATA SIZE

		P	R	F	Test Accuracy	CV Accuracy	Training Accuracy
Supervised (80% train)	0	0.63	0.67	0.65	76.72%	76.50%	99.22%
	1	0.85	0.85	0.85			
	2	0.49	0.44	0.46			
15% train	0	0.60	0.63	0.61	76.27%	96.48%	99.94%
	1	0.83	0.87	0.85			
	2	0.47	0.23	0.31			
20% train	0	0.63	0.66	0.65	78.27%	94.82%	99.84%
	1	0.85	0.87	0.86			
	2	0.55	0.34	0.42			
25% train	0	0.63	0.66	0.65	78.05%	92.27%	99.79%
	1	0.86	0.86	0.86			
	2	0.52	0.40	0.45			

V. DISCUSSION

Our semi-supervised self-training model has shown to be nearly as effective as our supervised learning model. This shows that similar high-quality results can be produced with significantly less training data required compared to supervised learning. This will cut down on the use of experts' time required to prepare the dataset because there are fewer data to be labeled to produce the same quality results or potentially even better.

After looking at the test data and comparing the actual label with the predicted label of each tweet, we noticed a pattern when the model incorrectly predicted the test data. The pattern we noticed when tweets were mislabeled was the occurrence of swears. If there was a swear the algorithm predicted that the tweet was Hate. For example, this tweet: "BRO this is how the fucking common cold works, this is not scary." The tweet was classified as Hate but should be considered Neutral. Other mislabeling occurred because swears were used in tweets that are labeled as counter-hate.

For example, “The virus is isn't Chinese. It is a coronavirus. You are a both a racist and feckin' arsehole. You did nothing and now that it spreads you scream racist crap. Scalegoating is what nazis do. Stop being a nazi.” This shows that our model is not good at deciphering between when a swear is used as hate speech, defensively, or neither.

We also noticed the amount of incorrectly predicted labels in the test data slightly increased as the training size grew.

One thing in our experiment that could have improved the classification results and refined training data would be to have multiple people label the same tweet. In our dataset we had three people label three different portions of the data that was used. The first portion was labeled by the creator of the dataset, and two student researchers labeled the other two portions. We were each given two different sets of tweets to label and then our individual sets were added to the previous one creating our current dataset. Instead, we could have all labeled the same tweet so there would be multiple labels attached to one tweet. If there was greater than a 50% agreement on what the tweet should be labeled, then it would be added to our dataset. This would help with consistent labeling and minimize bias by having more than one expert label the data.

Lastly, when comparing the accuracies of each trial we can see a slight overfitting issue. The training accuracy is slightly higher than the cross-validated accuracy. As the training data size increases so do the difference between the two accuracies. Although there is slight overfitting in our self-training algorithm, there is a larger amount of overfitting in the supervised learning model. This indicates that as more training data is initially available the model becomes too closely fitted to the training data to generalize unseen data. Based on our results, our self-training model does a better job at combatting the overfitting issue than the supervised model.

VI. Conclusion

We experimented to see the effectiveness of semi-supervised learning and compared it to supervised learning. We evaluated the three different training data sizes in our self-training algorithm which included 15%, 20%, and 25% as our training data with a constant 10% testing size. We randomly shuffled the data each time and compared our results to supervised learning which had 80% training data and 20% test.

Overall, as the training data increases so do the F1 scores and the test accuracies. Based on our results, anywhere between 20% and 25% of training data is ideal because it produces high F1 scores and high accuracies.

Our model performs well given the difference in training data when compared to supervised learning. The test accuracy of 20% and 25% of training data are higher than the test accuracy of supervised learning. Also, the F1 scores of the 25% training data are nearly the same as the supervised learning F1 scores when there was 80% training data. This is particularly great considering our model has significantly fewer labeled data to train on and can produce similar results.

Future works would include testing other machine learning algorithms to compare with our results. We used self-training, but we could also use co-training to see if the result could improve. Lastly, we could try to improve the overfitting issue that occurs in our model.

VII. Acknowledgements

This work is partially sponsored by the National Security Agency under the grant H98230-22-1-0017, the National Science Foundation award 1719498, and the University of Tampa Office of Undergraduate Research and Inquiry. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the sponsor.

VIII. References

[1] Ziems, Caleb, et al. "Racism is a virus: Anti-asian hate and counterhate in social media during the covid-19 crisis." *arXiv preprint arXiv:2005.12423* (2020).

[2] Kim, J. Y., and A. Kesari. “Misinformation and Hate Speech: The Case of Anti-Asian Hate Speech During the COVID-19 Pandemic”. *Journal of Online Trust and Safety*, vol. 1, no. 1, Oct. 2021, doi:10.54501/jots.v1i1.13.

[3] A. Singh, N. Thakur and A. Sharma, "A review of supervised machine learning algorithms," *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 1310-1315.

[4] Breiman L, Friedman J, Stone C J, et al. Classification and regression trees [M]. Chapman& Hall/CRC, 1984.

[5] McClosky, David, Eugene Charniak, and Mark Johnson. "Effective self-training for parsing." *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. 2006.

[6] Blum, Avrim, and Tom Mitchell. "Combining labeled and unlabeled data with co-training." *Proceedings of the eleventh annual conference on Computational learning theory*. 1998.

[7] S. V. M. Vishwanathan and M. Narasimha Murty, "SSVM: a simple SVM algorithm," *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, 2002, pp. 2393-2398 vol.3, doi: 10.1109/IJCNN.2002.1007516.

[8] Sandeep Shah, Xiaohong Yuan, Zanetta Tyler, “An analysis of COVID-19 related Twitter data for Asian hate speech using machine learning algorithms”, The 1st International Conference on AI in Cybersecurity (ICAIC), May 24-26, 2022.

[9] Davidson, Thomas, et al. "Automated hate speech detection and the problem of offensive language." *Proceedings of the international AAAI conference on web and social media*. Vol. 11. No. 1. 2017.

[10] S. Alsafari and S. Sadaoui, "Semi-Supervised Self-Learning for Arabic Hate Speech Detection," 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021, pp. 863-868, doi: 10.1109/SMC52423.2021.9659134.

[11] L. Gao, S. Shah, N. Assery, X. Yuan, X. Qu and K. Roy, "Semi-Supervised Self Training to Assess the Credibility of Tweets," 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2021, pp. 1532-1537, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00206.