Machines that Resent Hate Speech
Using Machine Learning to Detect and Respond to Hate Speech

Stephen Sigman, Daniel Soares, Ely Merenstein

Abstract

The widespread prevalence of social media in recent years has made hate speech detection increasingly important. This necessity for detection has become even more relevant within the last two months as Facebook has come under fire for their lack of limitations on user hate speech. One of the challenges with online hate speech is not only detecting it, but effectively responding in a way that might prevent the hate from escalating. Expanding on the hate speech detection model created by Qian et al., we use a Stochastic Gradient Descent model to detect hateful messages with reasonable accuracy. From there, we create two text generation models using the Python Textgenrnn and Sequence-to-Sequence libraries, both designed to generate responses given a specific, hateful comment. We evaluate our system using hate speech comments and responses collected from Reddit and Gab.

Introduction

As the global political climate becomes increasingly polarized, social media can magnify discord. With increasing numbers of people having a presence on social media, individuals with racist, misogynistic, or homophobic viewpoints, among other hateful opinions, are able to find online communities that reinforce their beliefs. Alarmingly, 67% of Americans agree that people should be able to make public statements that are offensive to minority groups (Poushter, 2015). The links between online hate speech and acts of violence are numerous and growing (Laub, 2019). However, it is unclear whether current regulations across the international community are able to address the issue of hate speech effectively in the face of those who want to speak unrestricted (O'Regan, 2018). Therefore, it is vital that governments and social media companies ask the question: how do you meaningfully reduce the spread of online hate speech while also mitigating concerns of stifling free speech?

Computer scientists have tasked themselves with constructing approaches to detect and prevent hate speech through education and inclusion rather than censorship and silencing. We seek to corroborate the approach of Qian et al. (2019) in the twofold task of detecting hate speech in a comment and then generating a constructive response to said hateful comment. The generation of an interventional response, rather than simply blocking posts and suspending offending users, is important because the most effective way to de-radicalize individuals with hateful viewpoints is to "change what people think instead of merely changing what they do" (Qian et al., 2019). We use the same dataset as Qian et al.: a fully-labeled collection of 5K conversations retrieved from Reddit, and 12K conversations retrieved from Gab, along with manually written responses to conversations containing hate speech. This data is used to train models to detect hate speech and then use the context of the conversation and pre-written responses to automatically generate an appropriate response to a hateful comment. The most popular model to classify hate speech is the support vector machine, which we employ as well. There is new research also investigating the viability of using deep learning-based models to identify hate speech (Zhang & Luo, 2018). However, there are two significant distinctions introduced by Qian et al. that we hope to corroborate: including conversational context rather than individual posts, and generating responses to hateful posts. In short, existing social theories say that constructively engaging with hate speech leads to better outcomes than simply blocking hate speech. By keeping these theories in mind during our research rather than focusing solely on models for detecting hate speech in isolated comments, we go one step closer to a real-world application.

System Design

Our system uses three different machine learning models. For the first half of the system, we use Scikit Learn's stochastic gradient descent model to detect if a given text contains hate speech. Gradient descent algorithms are commonly used in machine learning due to their simplicity and efficacy. The algorithm is a function that trains a support vector machine on given data until it converges on a minimum slope (derivative). Stochastic gradient descent optimizes the gradient descent algorithm by randomly jumping between points, helping to prevent the algorithm from being stuck in a local minimum, as seen in Figure 1.

For posts in which hate speech is detected, the second half of our system generates a response in an attempt to de-escalate the situation. For our text generation we use two different models: an implementation of Textgenrnn, which gen-

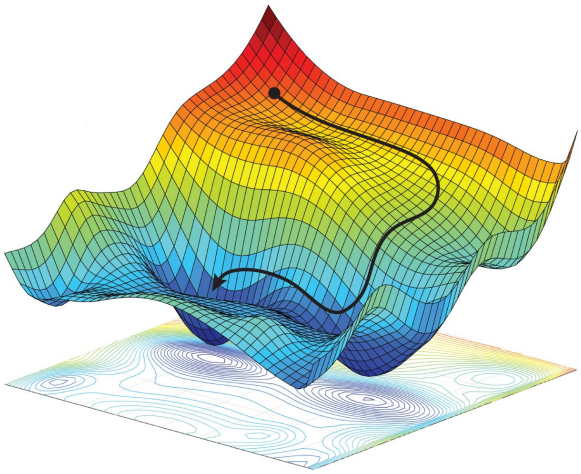

Figure 1: Stochastic Gradient Descent: a visual representation of the concept (Azizan and Hassibi, 2018).

erates a generic response, and a Seq2Seq model which uses the content of the hate speech to generate a reply catered to the specific context.

Textgenrnn

Textgenrnn is a model trained on top of Keras/Tensorflow that can generate text after being trained on an input text file (Woolf, 2017/2020). Because of the generic nature of many of the responses documented in Qian et al., beginning with a basic text generator is an appropriate first step to learn how to generate responses to hate speech. Figure 2 depicts the layers of the RNN on default parameters. After a sequence (here, 40 characters long) is embedded into a 100-dimensional character embedding vector, it is then fed into two LSTM recurrent layers. The Attention layer weighs the most important temporal features and averages them together, which is then mapped to probabilities for the next character in the sequence.

Seq2Seq

The sequence-to-sequence (Seq2Seq) implementation we use as our second means of text generation is nearly the exact model Chollet uses (2017). In this implementation, a LSTM network called the encoder processes the tokenized input string. While the actual output of this network is ignored, the resulting states are captured and used as the initial states in a second LSTM network, the decoder. The decoder receives as input a token sequence, initially containing only a "start token" (represented as <GO> in Figure 3). It then predicts the most likely subsequent token in the output string and appends it to the token sequence. Some or all of the token sequence (in our implementation, just the most recent token) is then used as input in the following decoding iteration. The decoder continues to loop as such until it reaches a set maximum length of tokens or until the decoder outputs a "stop token" (Chollet, 2017). Our LSTM networks contain input layers with 100 nodes.

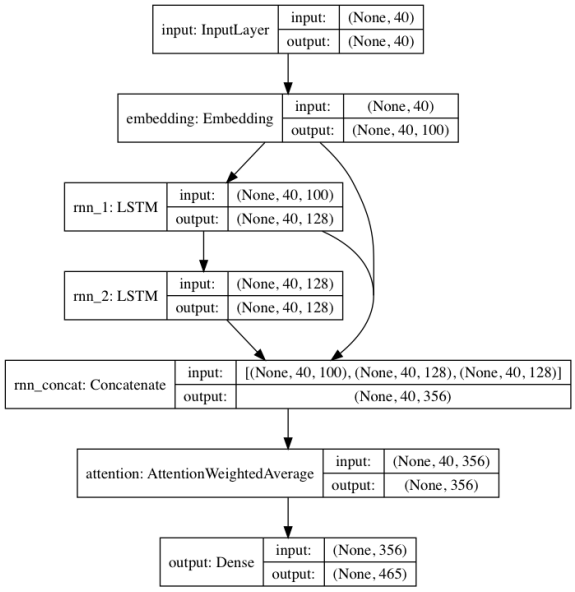

Figure 2: A visual representation of the Textgenrnn model (Minimaxir).

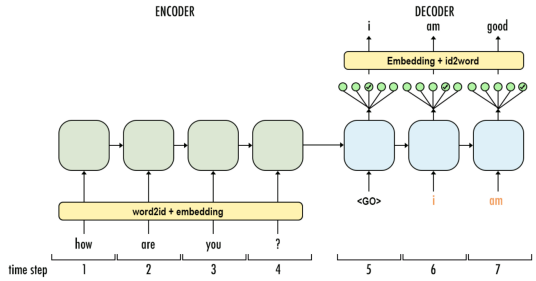

Figure 3: A visual representation of a Seq2Seq model (Long, 2017).

Dataset

The data for this study comes from the dataset Qian et al. use, entitled, "A Benchmark Dataset for Learning to Intervene in Online Hate Speech" (2019). The dataset consists of 5,020 conversation threads from Reddit and 11,825 conversation threads from Gab, each consisting of between one and twenty posts, with a total of 22,309[1] Reddit posts and 33,776 Gab posts. Following each conversation in the dataset is a list of the indices of posts in that conversation, if any, that are marked as hate speech. For conversations that do include hate speech, the dataset also provides three responses from Amazon Mechanical Turk workers that call out or discourage the hateful aspects of the posts.

[1]Qian et al. (2019) note 22,324 Reddit posts. The 15-post difference between their count and ours may be due to differences in the efficacies of their and our respective programs for splitting conversations into posts. For example, it took us some time to discover that splitting Reddit conversations into posts by the newline character was insufficient due to the unescaped presence of this character in a few posts; there may be similar matters we have yet to discover.

Both the Gab and Reddit datasets contain horrible examples of hate speech, but on the whole the comments from Gab seem to be significantly more hateful and racist. Some examples of comments from Reddit are: "this is the most r****ded thread I've seen in a while," "a subsection of r****ded Hungarians," and "dumb f***ing hillbilly white people," while comments from Gab generally look like: "you sound like a f**got," "you can't even spell it n**ger," and "you are just a stupid n**ger lover."

The responses from the dataset ranged from ones that were generic and reusable to ones that specifically addressed particularly offensive words. Some examples of the generic responses are: "please don't use such offensive language," "your comment is completely unhelpful and only shows your hatred," "the language in this post is completely incorrect and hateful," and "if you want to be taken seriously you need to find a less offensive way to get your point across." An example of a more targeted response is: "please don't use the word n**ger, it's offensive to a lot of people not to mention all the pain that word has caused over the years."

Experiments

Classification

For our hate speech classifier we decided to use the Scikit Learn suite of machine learning models. We chose to compare three different types of models to discover which would be most compatible with our data set: SVC, LinearSVC, and SGDClassifier. In an effort to identify the model which would yield the highest accuracy, we created six instances of each type of model, each with different hyperparameters. Ultimately, we were able to obtain the greatest accuracy using an SGDClassifier with default parameters (by default, the SGDClassigier uses a hinge loss function). Upon the completion of preprocessing, we split the dataset into a training and test set. Afterwards, we split the training data again into a training and validation set. The training set was used to train our model, which was then validated for proper fitting using our validation set. Finally, our classification model was tested for accuracy using our test data set.

Response Generation

We ran two separate response generation experiments: one using the Textgenrnn library and one involving a Sequence-to-Sequence procedure implementation.

Textgenrnn. To preserve computational power and time, the Textgenrnn model was run on 500 lines of responses from the Gab dataset. By default, Textgenrnn processes sequences at character level, requiring less overall data than processing sequences word-by-word. The key goal of employing Textgenrnn is to investigate if reasonable responses can be generated using a relatively simple model. Our model contains two recurrent LSTM layers consisting of 128 nodes each. On a 2.4 GHz Dual-Core Intel Core i5 processor, 10 epochs of training takes approximately 30 minutes. The model employs the "categorical_crossentropy" loss function because each predicted character in a sequence has a wide range of possibilities.

Seq2Seq. Experimentation with Seq2Seq began with the input of very small amounts of data from the dataset—anywhere between 5 and 100 conversations—into the model, which was trained with a "categorical_crossentropy" loss function for 10-50 epochs, using a batch-size of 10 and outputting loss on a validation set that consisted of 20% of the total input data. Initially, tokens used for input parsing and output generation were individual characters; we abandoned this approach quickly after the model failed to generate any English words. However, even using complete words as tokens, the model failed to generate any remotely logical series of tokens at this stage. Furthermore, any given time we ran the model at this stage, all strings submitted as input to the model yielded precisely the same output as each other. These issues implied either a need for more data and more experimentation with hyperparameter values, or a fundamentally incorrect aspect of the structure of the model.

To check whether or not the fault lay within the way the model was built, we created a basic dataset using the letters 'a' through 'e' as input tokens that mapped respectively and directly to the letters 'b' through 'f' in output strings. We trained the model on no more than 10 strings consisting of these tokens, then tested the model's ability to generate the expected outputs of both the same strings used as training data and a few new strings constructed of the same tokens. While the output was not always exactly as expected, certain runs of the program did yield the exact output expected for some inputs based on the mapping of the letters. Additionally, any given run of the program produced multiple distinct output strings.

Thus reassured of the presence of some degree of artificial intelligence in our current Seq2Seq implementation, we returned to the dataset of conversations. After we increased the learning rate beyond the default value (0.001) for the Keras RMSprop classifier, the model yielded logical sequences of tokens. Given this, we set up a grid of hyperparameters to test: we tested learning rates of 0.01 and 0.005 and epoch counts of 40, 60, and 100. We were unable to load the complete dataset into memory at this stage and therefore ran the above hyperparameter tests separately on 500 Gab conversations and 500 Reddit conversations. For each hyperparameter-subdataset combination, we submitted 100 of the training posts to the trained model (to attain a simple indicator of the effectiveness of the model even before introducing unseen testing data) and had the program print the generated response. Using the default Keras verbose setting and a validation_split parameter value of 0.1 for fitting, we also were able to see the loss and validation loss after each epoch.

Results

Classification

After training the classification model, we were able to achieve an accuracy of 91.64% on our validation set and 91.33% on our test set. We ultimately were forced to exclude the training of our non-linear SVCs since they required unreasonably high training times given our time window.

Table 1: Seq2Seq Gab Response Generation Losses at Final Epoch (trained on first 500 conversations)

Epochs	loss, $\eta = 0.01$	val loss, $\eta = 0.01$	loss, $\eta = 0.005$	val loss, $\eta = 0.005$
40	0.2349	2.7811	0.2426	2.5019
60	0.2187	2.7507	0.2167	2.7323
100	0.2169	2.8227	0.2105	2.8871

Table 2: Seq2Seq Reddit Response Generation Losses at Final Epoch (trained on first 500 conversations)

Epochs	loss, $\eta = 0.01$	val loss, $\eta = 0.01$	loss, $\eta = 0.005$	val loss, $\eta = 0.005$
40	0.2494	2.2483	0.2528	2.1947
60	0.2359	2.3867	0.2226	2.2580
100	0.2248	2.5321	0.2181	2.5250

However, while testing the different classification models, we were able to tune each version to yield a minimum accuracy of 89%. While 91% is not a significant improvement from our lowest accuracy model, we feel that our chosen model has a satisfactory accuracy for simple hate speech detection.

Textgenrnn

After ten epochs of training, our textgenrnn model was able to generate reasonable responses to hate speech. Some of the responses generated mirror responses from the training data, which is to be expected because there may exist multiple instances of each response within the dataset. The reported loss after training is 0.3368, and the learning rate is 4.00×10^{-4}. While the response generation would likely be more robust if the model trained over the entire dataset, the fact that the model could reproduce some of the pre-written responses with limited training is a promising sign that text generation models built on a recurrent network could be highly effective in generating specific responses to hateful comments.

Seq2Seq

The final loss and validation loss for each epoch count/learning rate pairing are shown in Tables 1 and 2 for Gab data and Reddit data, respectively. Given initial losses of approximately 2.9 for the Gab set and approximately 3.2 for the Reddit set, the training losses in the tables are evidence of successful fitting over time. The relatively high validation losses, however, provide a less hopeful result in terms of the ability of the model to generalize to unseen data. Analysis of the changing validation losses over time show that validation loss for the Gab set was minimized (around 2.1) within 10 epochs, and for the Reddit set between 10 and 20 epochs (around 1.85), implying overfitting. However, because of the complexity of the problem of text generation, it is, at this stage in our study, worth studying the predictions of the model even after this many epochs. This is because, by evaluating the model on the training set after extensive overfitting, we allow the model to display its current maximum possible efficacy at text generation, even if the specific application (predicting the output of the training set) is trivial.

Tables 3 and 4 show, respectively for Gab and Reddit data, the distributions of responses generated for each epoch count/learning rate pairing. The responses shown in this table are all exact copies of tokenized versions of responses from the dataset. This demonstrates the ability of the model to learn logical, if overused, sequences of words, a phenomenon we witnessed with even smaller portions of the dataset given a sufficiently high learning rate. However, training on an entire 500 conversations also caused the model to generate multiple distinct (i.e. input-dependent) responses during the evaluations of certain epoch count/learning rate pairings; the model had been unable to do this on smaller subsets of the hate speech dataset. Of particular note is the response generated exactly once in the pairing (Epochs=100, η=0.01). Unlike some other responses, this exact response was provided only once in the entire dataset—for the same input post that prompted the generation of this response. This input post was the longest of any used to test the model in this epoch count/learning rate pairing—2,258 characters, more than 200 longer than the second-longest—and therefore stood out somewhat from the other input posts. The fact that the model identified this marks a promising start.

Discussion and Conclusion

Our model worked well for low-level hate speech detection. However, if we were to use our system in a production environment, our group would have to tune our model to a greater degree of accuracy, as missing 9% of all hate speech can easily become an immense problem given a large volume of comments. A main reason for our unsatisfactory accuracy was due to our constraints on time to make and train our model, as well as our limited computing power. Given more time or faster computers, we may have been able to produce a model of greater accuracy using either a non-linear SVC or a different set of hyperparameters.

That said, attaining 91% accuracy relatively quickly and simply—as well as nearly identical accuracy on much smaller subsets of the dataset—surprised us somewhat. After some investigation of the dataset, we discovered that many posts, particularly in the Gab dataset, are labeled as hate speech if and only if they contain a derogatory word. If this is indeed a consistent pattern throughout the dataset, it

Table 3: Seq2Seq Gab Responses (trained on first 500 conversations)

Epochs	$\eta = 0.01$	$\eta = 0.005$
40	"use of the r word is unacceptable in discourse as it demeans and insults people with mental disabilities " (100%)	"use of the r word is unacceptable in discourse as it demeans and insults people with mental disabilities " (100%)
60	"use of the n word is unacceptable in discourse as it demeans and insults blacks " (83%) "please do not use offensive slur s " (7%) "use of the slur here is unacceptable in our discourse as it demeans and insults gays " (10%)	"use of the slurs here is unacceptable in our discourse as it demeans and insults and alienates others " (100%)
100	"use of the r word is unacceptable in our discourse as it demeans and insults people with mental disabilities " (100%)	"use of the slur here is unacceptable in our discourse as it demeans and insults gays " (100%)

means that a relatively few select features of the post vectors determine the output. Given this, it would be interesting to see if our model trains and performs well on other datasets where hate speech is labeled as such for more subtle reasons.

In the future, our responses could be improved by having our model look at all the comments in a given thread, to gain a deeper understanding of the context. For example, one of the hate comments in the Gab dataset is the phrase "dumb c*nt." While the system could give a reasonable response to the given comment, perhaps it would be more effective if the system actually knew to whom the user was referring so that it could give a more targeted de-escalation effort.

While our research takes the approach that it is preferable to constructively intervene in communities with frequent hate speech, that does not disqualify the fact that it is sometimes necessary to ban accounts or entire communities outright. Reddit, as a part of a major expansion of its rules, banned more than 2,000 communities including r/The_Donald, which is sampled in our dataset (Newton, 2020). We hope that continued exploration into our research direction will give social media platforms an additional course of action for proactively mitigating hate speech so that communities may remain active, rather than having to reactively ban communities that have become too toxic to effectively moderate.

References

Azizan, N., & Hassibi, B. (2020, January 2). Stochastic Gradient/Mirror Descent: Minimax Optimality and Implicit Regularization. http://www.its.caltech.edu/ nazizanr/papers/SMD.html

Chollet, F. (2017, September 29). A ten-minute introduction to sequence-to-sequence learning in Keras. https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html

Laub, Z. (2019, June 7). Hate Speech on Social Media: Global Comparisons. Council on Foreign Relations. https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons

Long. (2017, July 9). Seq2Seq. Medium. https://medium.com/@Aj.Cheng/seq2seq-18a0730d1d77

Newton, C. (2020, June 29). Reddit bans r/The_Donald and r/ChapoTrapHouse as part of a major expansion of its rules. The Verge. https://www.theverge.com/2020/6/29/21304947/reddit-ban-subreddits-the-donald-chapo-trap-house-new-content-policy-rules

O'Regan, C. (2018). Hate Speech Online: An (Intractable) Contemporary Challenge? Current Legal Problems, 71(1), 403–429. https://doi.org/10.1093/clp/cuy012

Poushter, J. (2015, November 20). 40% of Millennials OK with limiting speech offensive to minorities. Pew Research Center. https://www.pewresearch.org/fact-tank/2015/11/20/40-of-millennials-ok-with-limiting-speech-offensive-to-minorities/

Qian, J. (2020). Jing-qian/A-Benchmark-Dataset-for-Learning-to-Intervene-in-Online-Hate-Speech. https://github.com/jing-qian/A-Benchmark-Dataset-for-Learning-to-Intervene-in-Online-Hate-Speech (Original work published 2019)

Qian, J., Bethke, A., Liu, Y., Belding, E., & Wang, W. Y. (2019). A Benchmark Dataset for Learning to Intervene in Online Hate Speech. ArXiv:1909.04251 [Cs]. http://arxiv.org/abs/1909.04251

Woolf, M. (2020). Minimaxir/textgenrnn [Python]. https://github.com/minimaxir/textgenrnn (Original work published 2017)

Zhang, Z., & Luo, L. (2018). Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter. ArXiv:1803.03662 [Cs]. http://arxiv.org/abs/1803.03662

Table 4: Seq2Seq Reddit Responses (trained on first 500 conversations)

Epochs	$\eta = 0.01$	$\eta = 0.005$
40	"using the word c*nt is offensive as its a direct attack to someone based on their gender refrain from such words " (100%)	"please refrain from using hateful ableist language in your posts it adds nothing of value to the discussion in this thread " (100%)
60	"please refrain from using hateful sexist language in your posts it adds nothing of value to your argument or the discussion in this thread " (100%)	"using the word c*nt is offensive as its a direct attack to someone based on their gender refrain from such words " (47%) "please refrain from using hateful sexist language in your posts it adds nothing productive to the conversation of the sub " (53%)
100	"please refrain from the use of hateful ableist language in your posts it doesn t help your argument or add anything of value to the thread " (99%) "dont use that r word as a catch all i thought we were done with that " (1%)	"please refrain from using hateful sexist language in your posts it adds nothing of value to the conversation in this thread " (100%)