Toksik: Employing Machine Learning to Detect Toxicity in Online Posts

Pratham Gandhi

***Abstract*— An experimental development approach was created and implemented which enabled the training, testing, and tweaking of three separate machine learning classification models Naïve Bayes, Neural Network, and Support Vector Machine in order to observe differences in accuracy and prediction time when predicting and identifying toxic (racist, sexist, or otherwise offensive or abusive) text in online posts. The models were trained in different scenarios using different parts of the training data set in order to observe weaknesses and strengths of each of the models and determine the applicability of the models to various use-cases in industry. Ultimately, a Bayes classifier proved to be most effective, reaching an accuracy of 91.97% on more than 170,000 phrase testing set from Google.**

I. INTRODUCTION

As the use of social media platforms is growing at the fastest rate in history, so too is the amount of hurtful content posted on these platforms. Some of the largest social media companies on the planet are spending upwards of hundreds of millions of dollars each year on project teams and employees whose sole task is to combat and censor toxicity in posts on their platform. Furthermore, the applications which these project teams create are extremely limited in their effectiveness and applicability to diverse situations, as they are currently formulated to look for specific words or phrases and then flag the post. In the current implementation of these systems, a flagged post is then passed to both primary and secondary reviewers, humans, who must determine whether or not a given post violates the community guidelines or rules of the platform. This entire process is extremely resource intensive both in terms of monetary and human capital. An automated system which can perform at a higher accuracy level than humans will enable large social media companies to save untold sums of money and allocate these trained professionals, who are currently spending countless hours sifting through content, to more helpful positions.

This project explores potential applications of various machine learning models to flag a string of text as toxic, meaning it contains racism, sexism, hate speech, and other offensive or abusive language which might be hurtful to others. Additionally, the project aims to present a tested approach which can effectively identify these comments or posts. As machine learning technologies grow and evolve, we are now able to more closely model the human mind and its rational decision making capability. This brings forth the opportunity to use machine learning to emulate an effective substitute for those human employees. Additionally, a learning-based approach to this problem provides abstractability and generality which allows content reviewers to apply their experiences and knowledge to guide and train a predictive model. This project spans many subsets of science but primarily utilizes aspects of behavioral psychology, data science, and computer science.

II. METHODS

The first step to attaining a machine learning model which can provide accurate classification given a set of input features is to identify the features which are relevant to the classification. Additionally, all the data which is used to train the model must have values for each of the features which are used in the classification. Another important item to consider is that different types of machine learning models will require different types of input features; some require floating point numbers, some require string text, and some require binary (0 or 1) inputs. The classification model which would work best could not be determined prior to constructing the classification pipeline, in this case, so the features were chosen such that they could be applied to a range of classification models. It was determined that the optimal way in which to represent the strings of text which were to be classified was to break them down into features using the Bag of Words approach. In this approach, every time the classifier comes across a new word in the training data set, it assigns the word a unique number ID. A sentence or phrase can thus be represented as an array with the unique IDs of each word in the phrase and the number of occurrences of that word. This numerical representation of phrases is matched with the classification provided to the model in the training data set, and the model uses this to establish cause and effect relationships between the presence, prominence, and ordering of certain words in a phrase and the classification that phrase warrants. In order to ensure that the model understands that words such as the and of are not significant in providing a phrase with the meaning, and thus toxicity, which it carries, a term frequency-inverse document frequency weight was implemented. This weight considers the number of times a word shows up in a phrase and the number of times it shows up in the whole set of documents (the collection of phrases used for training the model) in order to assign that particular word this weight, which is proportional to the importance the word has in the phrase. This weight is calculated in the following manner:

$$\text{TF-IDF}(t, d, D) = f_{t|d} * \log(\frac{N}{o \in D : t \in o})$$

Where t is a given term in document d, trained on a data set D with N documents.

The classifications which the model aimed to be able to compute were whether or not a string of text was 'obscene,'

'threatening,' 'insulting,' or 'identity based hate.' Three machine learning models lend themselves best to the categorical classification required for this project: Naïve Bayes, Neural Networks, and Support Vector Machines. Naïve Bayes uses the following generalized approach to calculate the probability that a certain phrase, represented as features x in the form of the Bag of Words model, receives a given classification c:

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

Neural Networks are sets of neurons modeled by mathematical functions, whose outputs propagate through layers of other neurons for which they are inputs. The output of a given neuron with inputs x, each with weight w, and an overall correction factor b is represented as follows:

$$\text{Output} = 1/[1 + \exp(-\sum(x_i * w_i + b))]$$

Finally, support vector machines are classification models which calculate an $n+1$ dimensional vector which separates a vector space populated with n-dimensional feature vectors into respective classifications such that classification error is minimized. In many simple problems, traditional multi-dimensional optimization algorithms can be used to achieve this.

Each classification model has its own distinct advantages and was tested using a set of training data with 253,000 phrases compiled by Jigsaw and Google, both subsidiaries of Alphabet, in order to determine which was best for this application. This data set was reliable to use in training the model and didn't produce any outstanding errors in classification due to bias in the data set as, before they were added to the data set, all phrases were vetted by several people in both companies. After training each of the three chosen classifiers using the training data, they were tested on a 173,000 phrase training data set, also sourced from Google, and the accuracy of that particular model was analyzed.

After determining the model with the highest accuracy and adaptability based on the parameters and procedures described below, the model's trained state was broken down and stored in a 'pickle' file, using Python's pickle library. This pickle file can be downloaded, unpacked, and deployed onto a server for immediate use, or the model stored in this file can be retrained using the collection of data from the company which wants to implement this model.

III. Results

As described earlier, the three models were tested and the overall accuracy of each of them was recorded. In addition to recording the overall accuracy, the accuracy of each model for different situations in the training process (i.e. number of training phrases or number of words in training phrase) was measured in order to make conclusions about which models might be better suited to this application in the future, where there is more training data available or the types of use-cases tend towards longer strings of the input text. The Bayes Classifier was able to achieve a maximum 91.97% overall accuracy for classifying general toxicity throughout the testing set from Google. The other classifiers, namely Neural Network and Support Vector Machine, achieved 67.91% and 90.89% overall accuracy, respectively.

Phrases with different word counts were tested in order to investigate how much each model relied on the context of the rest of the words in a phrase in order to provide an accurate classification.

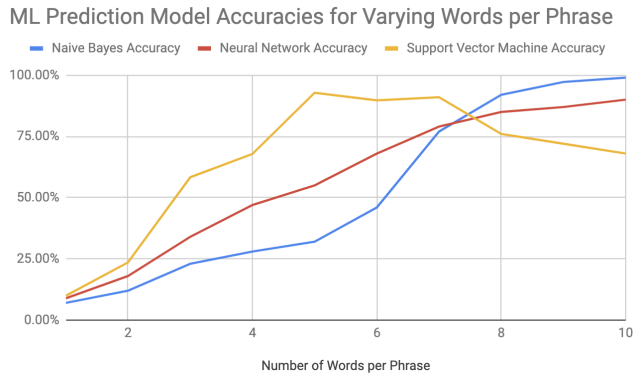

Fig. 1. Each classification model's average accuracy on predicting the toxicity of all test phrases of 1-10 words, stratified by word count, was recorded after training on the same full training data set, in order to observe dependence on input characteristics, indicative of breadth of applicability.

Different sizes of training data sets were randomly sampled from the entire corpus of training phrases in order to investigate the effectiveness of the classifiers with lower or higher volumes of training data. This is important to investigate because, in real-world applications of these models, large social media companies will have several orders of magnitude more training data than what was available during this experiment, whereas smaller startups looking to implement this technology will have extremely limited amounts of training data.

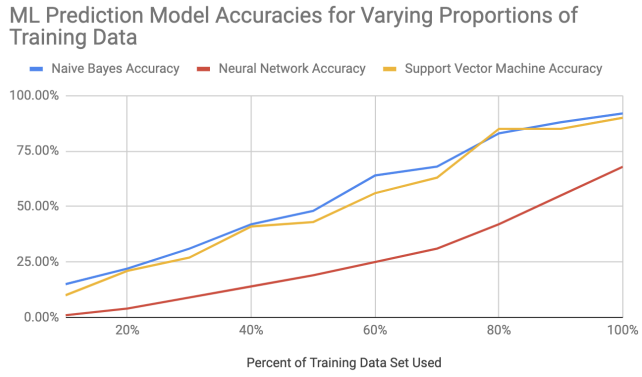

Fig. 2. Each classification model was trained using varying proportions of the the training data set (randomly selected) and then average accuracy was measured on the entire testing data set, in order to observe dependence on training data.

IV. Discussion

From a simple preliminary analysis, it can be seen that, as expected, the majority of the approximately one hundred seventy thousand comments in the testing data set were not toxic of any kind. However, the comments which were indeed toxic yielded an interesting pattern, which can be seen in the graph below:

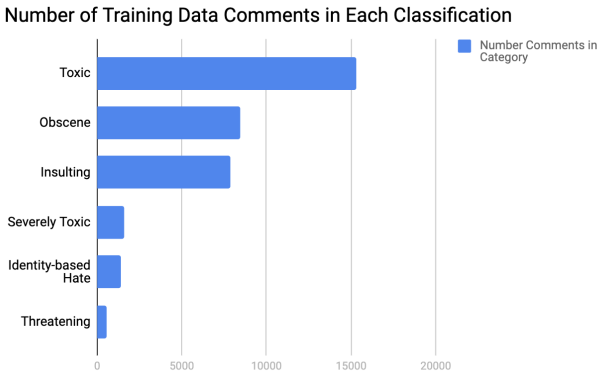

Fig. 3. The training data was broken up by classification in order to observe weaknesses in training coverage.

Note that one comment can be classified as representing more than one type of toxicity, and all of the comments in this graph are classified as toxic. Leading by a gap of over about 6,000 data points, the two biggest categories in which comments were classified, besides baseline toxic, were Obscene and Insulting. This likely resulted in a trained classifier which was much weaker in predicting severe toxicity, identity-based hate, and threats. This hypothesis could not be verified, though, as the testing data was not labeled as such.

With the amount of training data available during this project and the testing data used to evaluate each model, the Bayes classifier performed better as a whole. However, as can be seen from Figure 2, the accuracy of a Neural Network classifier grows nearly exponentially with the amount of training data the classification model is provided. This means that in industry-level applications, a Neural Network would likely outperform either of the other two classification models tested, as much more training data will be available for large companies to use to train their prediction model.

Additionally, for applications where lots more context is available in the form of longer posts, the analysis presented in Figure 1 clearly demonstrates that Naïve Bayes will be the most accurate prediction model, as expected. However, when shorter text blurbs are the only type of data available for training and classification, for example in the case of Snapchat or Twitter, Support Vector Machines will be the most accurate classification model.

V. Conclusion

There are now 4.3 billion internet users in the world. Approximately 80% of this population use some type of social media platform on a regular basis. If used properly, social media is very beneficial to our society. Major news outlets, corporations, and influential individuals use social media to deliver messages to the masses and keep the public informed in real time. However, this accessibility also opens doors to hurtful content spreading at an alarming rate. This study has clearly demonstrated that machine learning and natural language processing are important tools which can effectively be harnessed in order to combat this content. Specifically, for medium to large scale social media platforms with lots of data available to them, a Bayes classifier or Neural Network can be effectively implemented in order to classify toxicity, and at smaller scales, Support Vector Machines can perform the same classification with similar accuracy.

References

[1] Rebecca Bruce and Janyce Wiebe. Word-sense disambiguation using decomposable models. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pages 139146. Association for Computational Linguistics, 1994.

[2] Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. Computational intelligence, 22(2):110125, 2006.

[3] Yelena Mejova. Sentiment analysis: An overview. University of Iowa, Computer Science Department, 2009.

[4] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In Proceedings of the 2004 conference on empirical methods in natural language processing, 2004.

[5] Deepa Seetharaman. Facebook throws more money at wiping out hate speech and bad actors, May 2018.

[6] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation, 28(1):1121, 1972.

[7] Casey Whitelaw, Navendu Garg, and Shlomo Argamon. Using appraisal groups for sentiment analysis. In Proceedings of the 14th ACM international conference on Information and knowledge management, pages 625631. ACM, 2005.