An Analysis of Classification Models on a Toxic Comment Dataset

Debasmita Bhattacharya [1], Ruchika Dongre [1], Nikhil Saggi [2]

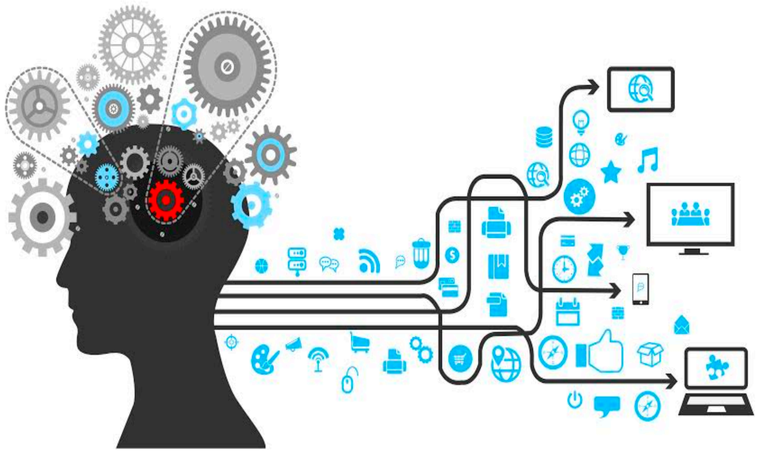
1. Computer Science, Cornell University, College of Engineering, Ithaca, N.Y.
2. Computer Science, Cornell University, College of Arts and Sciences, Ithaca, N.Y.

viewer discretion advised

INTRODUCTION

As the Internet becomes more anonymous, the need to moderate problematic and toxic comments increases. The diverse needs and varying degrees of online presence of companies that require this type of moderation means that each may prioritize different aspects of such detection. Some might need a highly efficient algorithm, while others care more about accuracy. Some may need to detect only the most severe comments, while others desire the most toxicity-free environment possible. To address these goals, we aim to analyze seven popular text classification models; these are namely the Gaussian Naive Bayes, Support Vector Machine, K Nearest Neighbors, Random Forest, Decision Tree, Logistic Regression, and BERT models. By applying these models to a Kaggle dataset composed of flagged Wikipedia comments, we plan to compare each model and its performance on detecting different types of toxicity such as threats, obscenity, insults, and identity-based hate. We hope that this work will not only serve as a supplement to existing models such as Google's Perspective, but also be of use to companies when deciding how best to ensure the online safety of their consumers.

CLASSIFICATION MODELS



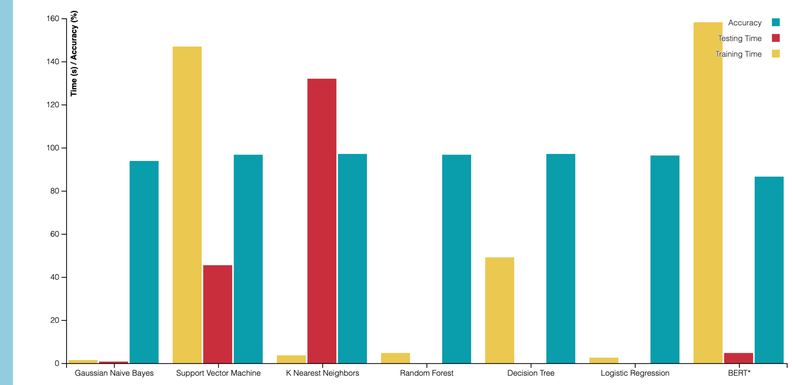
The tested classifiers range from well-known and tested models to newly developed models. Gaussian Naive Bayes is a standard classifier that exploits Bayesian probability by making assumptions of independent features to predict categories. K Nearest Neighbors considers proximity of data points when projected to a Euclidean plane of features to determine under which category each point falls. The Decision Tree model uses binary responses to guide inputs towards specific classifications. The Random Forest model combines multiple Decision Trees, each of which pointing to certain classifications, to incorporate greater variety and thus prevent overfitting. The Support Vector Machine model generates categories by partitioning data in a Euclidean plane of features. The Logistic Regression approach uses attributes to generate an equation that indicates an appropriate classification. And BERT is an NLP-oriented model that uses the context of words, combined with a predictive layer, to appropriately classify data.

MARKOV CHAINS

From RosettaCode, "a Markov chain algorithm determines the next most probable suffix word for a given prefix. To do this, a Markov chain program typically breaks an input text (training text) into a series of words, then by sliding along them in some fixed sized window, storing the first N words as a prefix and then the N + 1 word as a member of a set to choose from randomly for the suffix."
Our Markov chain generator is unique in that its vocabulary is taken only from the comments that our models classified as toxic. Thus, its "vocabulary" is only toxic words. Our program works by computing prediction vectors repeatedly, until we start to see some sort of stabilization – this means that the state vector has been achieved.
Then, we use this finalized prediction vector to get the next word! By continuing this process, we are able to generate full sentences and create a text generator from a dataset.

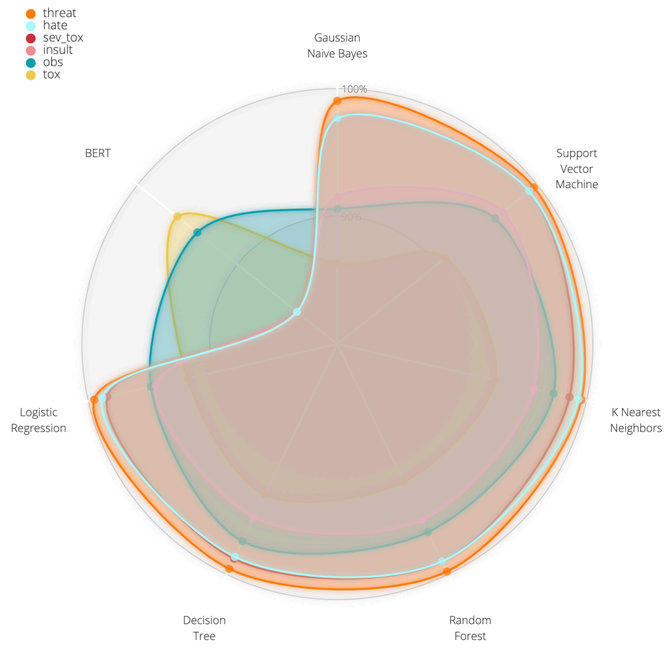
MODEL EFFICIENCIES

Our bar chart visualization compares the time taken to test and train each model in relation to accuracy of performance. All of our models are within 12% of each other in terms of accuracy, but the ones that are clearly the most efficient at training are Gaussian Naive Bayes, K Nearest Neighbors, Random Forest, and Logistic Regression. In terms of testing efficiency, Gaussian Naive Bayes, Random Forest, Decision Tree, and Logistic Regression perform the best. Overall, logistic regression seems to achieve the best balance between accuracy, training efficiency, and testing efficiency.



MODEL ACCURACIES

Our radar chart visualization compares the accuracy of each model per category of toxicity. As can be seen, the BERT classifier is the best at detecting the tox category; Gaussian Naive Bayes, decision trees, random forests are the best at detecting threats; logistic regression and K nearest neighbors are the best at detecting hate; support vector machine is the best at detecting insults; decision trees and K nearest neighbors are the best at detecting obscenity; and K nearest neighbors is the best at detecting severe toxicity. In general, most of the models perform uniformly and significantly well with regard to detecting hate and threats, and do far more poorly on detecting obscenity and toxicity.



OUR PROCESS

First, during the preprocessing stage, we standardized our data by removing all punctuation and converting every alphabetic character to lowercase, then testing for stop words and stem words. Then, in order to produce the highest possible accuracy, we decided to convert our words into numbers using three different vectorizers: CountVectorizer, HashingVectorizer, and TfidfVectorizer. Following an initial comparison of the performance of each of these, we found that TfidfVectorizer produced the best results, and so decided to exclusively use that vectorizer, discarding the other two. After that, we initialized each of our models with various settings (splitting them into training and testing sets) cross-validated. Finally, we fit and run our models. Based on the results of the previous steps, we were able to compare the time taken to test and train each model in relation to accuracy of performance, as well as the accuracy of each model per category of toxicity. These comparisons are exemplified in the following bar chart and radar chart visualizations respectively.

REAL-WORLD APPLICATIONS

The exponential growth of social media and community forums in recent years has revolutionized communication and content publishing. As people choose to interact more and more through online social networks, the anonymity of those platforms have contributed to the increase in propagation of hate speech and the organization of aggressive, hate-based activities, including hate crimes. According to a survey conducted by the Anti-Defamation League, 53% of Americans were subjected to hateful speech and harassment online in 2018. Of these respondents, 37% reported severe attacks, including sexual harassment and stalking. A third of Americans experienced online abuse targeting their sexual orientation, religion, race, ethnicity, gender identity, or disability. Overall, based on the results of a study carried out by the Pew Research Center, these statistics demonstrate a sharp increase of 18% in reported online toxicity since 2017. Thus, such online activity engenders the urgent need to effectively counter toxic comments in a virtual landscape that is not only difficult to monitor but also beyond the realm of traditional law enforcement.

Given the enormous scale of the Web and the millions of toxic posts -- in text-, image-, and video-format -- that circulate platforms like Facebook, Twitter, and YouTube, the detection and moderation of toxicity presents one of the most pressing challenges online. Measures such as hiring thousands of human moderators and training artificial intelligence software to crack down on abusive language have not yet solved the problem. In particular, moderators often misinterpret cultural cues and context due to the inherent subjectivity of human labelling, and algorithms continue to struggle to accurately interpret the meaning and intent of social media posts. In the face of these issues, 80% of Americans believe that the government should strengthen laws against online hate and harassment and improve training and resources for law enforcement. 75% of them want tech companies to make it easier to report toxic content and behavior, and 81% want social media to provide more ways of filtering out such content. Thus, it is essential that detection and moderation of online toxicity is carried out in a more efficient way than methods that are currently being employed. We hope that the results of our research provide insights into the possibilities for achieving such efficiency, and that the Internet becomes a safer space for everyone using it.

REFERENCES

Aken, B., Risch, J., Krestel, R. & Loser, A., (2018). Challenges for toxic comment classification: An in-depth error analysis. Beuth University of Applied Sciences.
-Used Wikipedia dataset.
-Raises concerns about out-of-vocabulary words, sentence dynamics (long-range dependencies), multi-word phrases.
-Most models have shortcomings that lead t o false negatives/positives with identifiable attributes (e.g. rhetorical questions, quotes of other comments)
Chakrabarty, N., (2019). A machine learning approach to comment toxicity classification. - Jalpaiguri Government Engineering College.
-Used Wikipedia dataset.
-Applied bag-of-words using word count vectorizer; set up term-document matrix; applied tf-idf.

ACKNOWLEDGEMENTS

Thank you to Prof. Jeff Rzeszotarski, the Cornell Data Science project team, the College of Arts and Sciences, and the College of Engineering for providing us with valuable feedback and the resources necessary to pursue our research.