HIGH SCHOOL EDITION
Journal of Student Research
Volume 11 Issue 2 (2022)

Comparison of the Efficacy of Natural Language Processing Algorithms at Classifying Cyberbullying Tweets

Eric Cui[1] and Christopher Brown[#]

[1]The Episcopal Academy, USA
[#]Advisor

ABSTRACT

Machine Learning is frequently used to predict and classify data. Natural Language Processing (NLP) uses machine learning to classify strings of words. There are many different machine learning models that can be used for NLP, with three main categories being regression, decision tree, and neural net models. Each has their own advantages and drawbacks. After being trained and tested on a set of tweets concerning cyberbullying, Logistic Regression, XGboost, and Long Short-Term Memory (LSTM) were compared in terms of several metrics, including accuracy, recall, precision, and f1-score. Afterwards, the metrics were considered in combination with model runtime and complexity to determine which model was most appropriate for the given dataset and other similar datasets. Logistic Regression was found to lack sufficient complexity to properly classify the data. LSTM had worse metrics than XGboost and had significantly higher complexity and runtime. XGboost performed best, with the highest metrics and relatively short runtime.

Introduction

Machine learning is a form of artificial intelligence that allows computers to make predictions about data without being explicitly programmed to do so. Supervised learning is a form of machine learning which uses pre-labeled data (or "training data) to create a model that predicts outcomes accurately. Natural Language Processing (NLP) is a form of supervised learning that specifically focuses on classifying sequences of human language. NLP is used largely in areas such as business analytics or social media as a method to classify text data, such as product reviews or social media posts, into categories with relatively high accuracy, saving time and manpower for humans needing to do the same tasks. [1]

There are three major types of algorithms used for NLP. The first is a regression model. A typical example is logistic regression, where the computer tries to fit the data points with a logistic curve that can subsequently be used to classify data. The second is a tree-based algorithm, which propagates slight variations of its "root node" through which data travels to reach a classification decision. An example of this is XGboost [2]. Finally, neural networks consist of a network of nodes which assigns different weights to various paths through which data can travel to predict the outcome. A type of neural network, Long Short-term Memory (LSTM), was used for this study.

Classification of tweets is a classic NLP problem since there is a large amount of data available and the problem is largely reflective of how social media companies process posts. Classifying tweets accurately could lead to various commercial applications such as targeted advertisement and identifying potential public safety threats such as tweets insinuating violence.

The objective of this study is to classify tweets according to their sentiments and evaluate the performance of various open-source models.

Related Work

Comparisons between different NLP algorithms have been studied extensively in the past [3-5]. However, generally the comparisons do not focus on the computing power required for each algorithm. Additionally, these studies generally do not compare algorithms in a way that puts emphasis on their type (regression/tree/neural net).

Falessi et al. (2010) [3] compared several different NLP techniques. However, this paper focused on machine learning in the context of benefiting human analysts and no tree-based algorithm was used.

Bakliwal et al. (2011) [4] compared an algorithm they developed with several other common NLP algorithms. However, no tree-based algorithms were used and the study used the common algorithms as benchmarks for the newly developed model.

Searle et al. (2020) [5] compared several algorithms including gradient boosting trees and deep learning approaches. However, the paper focuses on using algorithms such as gradient boosted trees to create better deep learning models.

Methodology

Dataset

In this study, a dataset of tweets [6] about cyberbullying was used. Each twitter was labeled either “not cyberbullying” or one form of cyberbullying, all of which were lumped into the category of “cyberbullying”. The dataset was chosen as a relatively simple yet applicable problem that could be used to gauge the efficacy of these three types of machine learning algorithms.

Preprocessing

Preprocessing of text data into a computer-readable format was necessary before applying each model. Each tweet was first tokenized into its constituent words using the nltk python package. Then, using a dictionary from the imdb package, each word was converted to a numerical value, with words not already in the dictionary being added. Each tweet was converted into a 1-D list of numbers.

Vectorization of data is the process of converting a string or list of numbers into computer-readable format. The most basic method of vectorization is one-hot encoding, a process where categorical variables are converted into indices in a vector, where all values of the vector are 0 except for the corresponding category, which has a value of 1

(Figure 1).

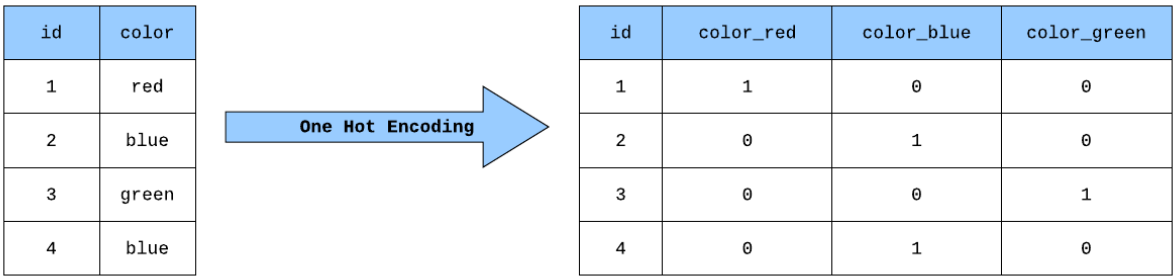

Figure 1: A demonstration of One-Hot Encoding using colors as categories

A different approach was taken for vectorization. The computer chooses a vector of specified size for each category, in this case, a unique vector that corresponds to each word. For example, the word "the" might correspond to [0.2, 0, -0.4, 0.5, 0, 0.9] when the vector size is set to 6. For the neural net approach, vectorization was performed by an embedding layer built into the keras package, while the tree approach used the TfidfVectorizer from the sklearn package.

Additionally, the dataset was split into training and validation sets. The algorithm first creates a model based on the training set, and is then tested on the validation set to see its effectiveness. In this study, a validation size of 0.25 was used.

Algorithms

Three algorithms were used for each of the primary NLP algorithm types. For regression, a logistic regression model was used, for tree-based XGboost, and for neural net LSTM. Since the focus of this paper is on comparisons between the three algorithms, only a brief overview of each algorithm is provided,

Logistic regression is a mathematical model that attempts to best fit a given set of data using a logistic curve. In the case of binary classification problems, the computer tries to fit a logistic curve to the data, and returns the value of the curve at a given value, corresponding to the probability of the value being "1". It is also the simplest of the three algorithms being used and requires the least computational power. This model was used as a baseline algorithm to compare the other more advanced algorithms to, since a simple logistic regression would not be expected to produce accurate results on a task with the complexity of NLP.

Tree-based algorithms function by first identifying which features best split the dataset (or the have the highest "information gain"), then repeatedly propagating nodes from this "root node" which each select their own features. In order to prevent overfitting of the data, the algorithm stops creating more nodes when the desired accuracy is achieved. Boosting is a concept based on the intuition that several weak classifiers, when combined, can create a much stronger classifier, and is often used with decision trees. Gradient boosted trees use several "weak learners" – generally smaller decision trees that capture some aspects of the dataset. Additionally, these weak learners each build off of the previous tree, ultimately creating a much more powerful gradient-boosted tree. Additionally, gradient-boosted algorithms do not require a fixed loss function, as they can operate with any easily differentiable function, allowing for further optimization. XGboost is an implementation of gradient boosting that is significantly faster and more accurate than traditional algorithms due to its capability to run parallel computations on one machine and its use of cross-

validation. It is the fastest of the three methods, but also sacrifices interpretability due to the large number of individual trees being used.

Neural networks are collections of interconnected nodes that closely resemble the human brain. At their core, neural networks function by changing the weights of the connections between nodes in order to minimize error. Recurrent Neural Networks (RNN) are a type of neural net that operate on sequential data, and are characterized by their "memory", which uses information from prior inputs to influence future ones. RNN's have "short-term memory", referring to the previously processed information within a single training cycle, and a "long-term memory", referring to the changes in weights between nodes between training cycles. LSTM's seek to create a short-term memory which is able to last for significantly more timesteps than a traditional RNN, making use of 1) a "forget gate", which filters non-important aspects of the long-term memory, 2) an "input gate", which filters the new input and decides which information to add, and 3) an "output gate", which determines the output of the cell using the long-term memory, the previous cell's output, and the new input (Figure 2). Though not as fast as XGboost, it provides more opportunity for further optimization and tuning, a subject not covered in this study. In this case, only a Dense layer (sigmoid) was added after the LSTM for the purpose of binary classification, and the only hyperparameter that was changed was the number of LSTM neurons.

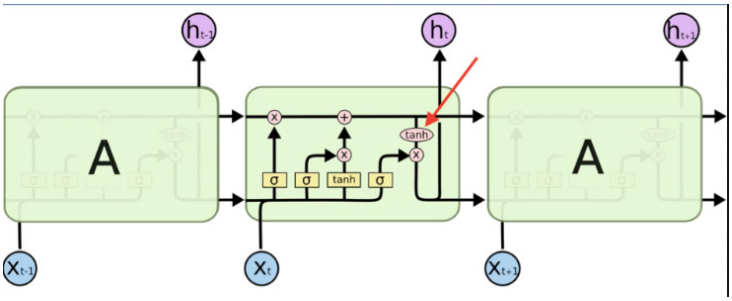

Figure 2: Three connected LSTM cells

Model Efficacy

The results are given in the form of accuracy, precision, recall, and f1-score for the validation set of each model, where values closer to 1 indicate a better model. TP/FP/TN/FN refer to True/False Positive/Negative

$$\text{Accuracy} = \frac{TP + TN}{TP+FP+TN+FN} \quad \text{Precision} = \frac{TP}{TP+FP} \quad \text{Recall} = \frac{TP}{TP+FN} \quad \text{F1} = \frac{2*precision*recall}{precision + recall}$$

Results

Accuracy, precision, recall, and F1 score was obtained from both training and validation sets. For most models, results from the training set is better than that from the validation set. However, Logistic Regression had no significant difference between the two because of its inability to properly classify data in the training set.

Hyperparameter tuning was performed for each model and the parameter combination which yields the best results is determined to be the final model. Results are presented from the final models.

Table 1: Summary of results on validation set

	Logistic Regression	LSTM	XGboost
Accuracy	0.659	0.814	0.860
Precision	0.355	0.677	0.867
Recall	0.076	0.774	0.980
F1 score	0.125	0.722	0.920

For the LSTM, the optimal number of neurons that yielded the greatest f1-score was 13.

Discussion

Expectations

The primary attribute of an effective model is its ability to correctly fit the training data without fitting it too closely. Overfitting is when the model too closely fits the training data and then is not able to extrapolate results. In general, more complex models can fit the training data better, but too much complexity results in overfitting.

Logistic Regression is by far the least complex of the three algorithms. Since it is not built to classify sequences and doesn't have a self-reinforcing element, it was expected to not fit the training set very well, meaning predictions on the validation set would not be accurate. Thus, Logistic Regression was predicted to have the lowest metrics of the three methods and would likely be scarcely better than a random guess.

LSTM was predicted to be the second-best algorithm. Although LSTM cells are built specifically for sequences, they often require large amounts of tuning in order to avoid overfitting. Since the LSTM neural network in this study was largely bare-bones, with only a single LSTM layer and one hyperparameter (number of LSTM cells), overfitting was expected. Because of this, LSTM was expected to be much better than Logistic Regression, but not as effective as XGboost.

XGboost was predicted to be the best algorithm. Since XGboost was a pre-installed package, the algorithm was more optimized than the other methods. The algorithm is sufficiently complex to accurately classify data, while also having features such as cross-validation that help to prevent overfitting. XGboost was expected to perform at a very high level.

Result Analysis

As can be seen from the results, logistic regression performed by far the worst, with the lowest F1 score. This is mostly attributed to the low recall, indicating that the model is missing a large number of positive values. The LSTM performed second-best, and has a higher recall compared to precision, indicating a higher rate of false positives. Similarly, XGboost had a low precision compared to recall, indicating the same issue as LSTM, but it was by far the most effective algorithm.

Algorithm efficacy

Algorithm efficacy goes beyond metrics. It is important to find which method has the highest accuracy, but practicality of each algorithm is equally important. Increased complexity may lead to higher accuracy, but also comes with the

drawback of increased necessity for computing power and longer runtime. It is generally favorable to select the simplest model that generates sufficient accuracy for the task at hand.

Logistic Regression takes the least computational power, but also is too inaccurate for any practical application. In general, LSTM runs the slowest, and requires vastly more computational power than tree-based methods. With proper tuning and a sufficiently large training set, it has potential to be more accurate than any other algorithm, but this further adds to runtime. For many practical applications where machine learning is simply used as the first filter (e.g. detecting cyberbullying in tweets), neural net runtime and computational strain are not worth it. Meanwhile, XGboost is not only more accurate than LSTM in most situations, but also runs significantly faster. For applications where small differences in accuracy do not have a significant impact, XGboost is a much more practical method. In very specific applications where every ounce of accuracy is required, a perfectly tuned LSTM would likely be more suitable.

In a commercial setting, training time often is more valuable than accuracy. For example, if a model is used to process tweets it needs to be able to keep up with the millions of tweets being sent every day. Slower neural net/deep learning approaches are preferred only in specific circumstances with a large static dataset.

Limitations

Only a single set of data was used to test the algorithms, meaning that biases in the data itself could have an effect on the results. Additionally, the data was not balanced, which could benefit some algorithms more than others. During preprocessing, no stop words were removed, which could potentially improve accuracy and runtime.

The LSTM was also limited by several factors. Firstly, minimal tuning was done to increase resistance to overfitting. Secondly, only 60 epochs were used in order to save runtime while also tuning the one hyperparameter. Finally, neural nets generally benefit from larger training sizes, so having more data could benefit the LSTM.

Future Study

Future studies primarily would improve upon the limitations outlined in the previous sections. Tuning the LSTM and changing more hyperparameters would be the primary extension to try and better compare neural nets and decision trees. Additionally, using multiple datasets and taking into account each algorithm's result for each would be another area for future study.

A future study could also look at different common NLP algorithms such as random forest, k-nearest neighbors, or Naive Bayes to conduct a more comprehensive review.

Analysis on the generalizability of each algorithm could also be a potential avenue for study. Processing tweets, product reviews, and dialogue are all vastly different tasks, and having an algorithm that is able to handle all of them could give it more merit. However, in some scenarios, a highly specialized algorithm would be better for very specific tasks.

Conclusion

Out of the three NLP methods of regression, neural net, and decision trees, neural nets and decision trees seem to hold the greatest promise. Regression is not able to handle the complexity of the problem. For smaller datasets, gradient-

boosted trees outperform imperfectly tuned neural networks. However, neural networks contain more potential for optimization, especially given larger datasets.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

1. Srivastava, A., Saini, S., & Gupta, D. (2019). Comparison of various machine learning techniques and ıts uses in different fields. In 2019 3rd International conference on electronics, communication and aerospace technology (ICECA) (pp. 81–86). Coimbatore, India.

2. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

3. Falessi, D., Cantone, G., & Canfora, G. (2010). A Comprehensive Characterization of NLP Techniques for Identifying Equivalent Requirements. Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. Παρουσιάστηκε στο Bolzano-Bozen, Italy. doi:10.1145/1852786.1852810

4. Bakliwal, A., Arora, P., Patil, A., & Varma, V. (2011, November). Towards Enhanced Opinion Classification using NLP Techniques. In Proceedings of the workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011) (pp. 101-107).

5. Searle, T., Ibrahim, Z., & Dobson, R. (2020). Comparing natural language processing techniques for Alzheimer's dementia prediction in spontaneous speech. arXiv preprint arXiv:2006.07358.

6. J. Wang, K. Fu, C.T. Lu, “SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection,” Proceedings of the 2020 IEEE International Conference on Big Data (IEEE BigData 2020), December 10-13, 2020.