CSC 240: Data Mining
University of Rochester
Final Project Report
Fall 2018

An Approach to Classifying Tweets

Training a Convolutional Neural Network to Identify Cyberbullying

By: Sammy Haq

Abstract

A 2-layer convolutional neural network is trained on hand-labeled cyberbullying tweets in order to determine whether it is possible to reliably classify tweets as harmful in nature or not.

After being cleaned of extraneous annotations such as apostrophes, the tweets are loaded as strings into a word2vec model to ensure that the semantics of words are taken into account. Then, the TF-IDF statistic is used to combine words into their previous sentences and fed into the convolutional neural network.

The convolutional neural network created is determined to have an average accuracy of 0.71.

Introduction

Interactions have taken to cyberspace. From major politicians such as the President of the United States to teenagers, the internet is the modern day wild west. Due to this widespread use, interactions that may have previously been in person can now be done online. This applies to workspace meetings such as the use of Slack, to social interactions like those on Facebook and Twitter.

Unfortunately, malicious interactions also take place on the internet, such as cyberbullying. Although these interactions can be policed by teachers and other authoritative figures in reality, policing these sort of interactions on the internet is almost non-existent. Furthermore, the internet is such an expansive place that the implementation of human moderators on these large forums is extremely expensive and ineffective. To aid moderators in their policing of cyberbullying, developing a filter to help automatically detect the more extreme cases of cyberbullying should be considered. This paper details an attempt to classify cyberbullying tweets by training a convolutional neural network on a hand-labeled Kaggle dataset by DataTurks. [1]

Methodology

For the purpose of this project, Python was chosen as the programming language -- this is due to the abundance of libraries that aid in machine learning projects, such as scikit-learn and Keras.

The scikit-learn library is a free machine learning library that is extremely simple and efficient to use. [2] Built on NumPy, SciPy, and matplotlib, this open source software can natively implement classifiers, regressions, and clusterers. This project mainly takes advantage of scikit-learn's dimensionality reduction, preprocessing and model selection libraries to process the tweets before creating the model.

The scikit-learn library, however, does not natively contain a classifier for convolutional neural networks -- Keras is instead used in order to accomplish this. An interface to the TensorFlow library, Keras is a library developed in order to facilitate easy experimentation with deep neural networks, such as convolutional neural networks. [3]

Sentences are not just composed of words -- the placement of these words next to each other imply semantic meaning. Instead of training the convolutional neural network on the tweet as a string type, word2vec is used to represent the semantics and placement of words in each tweet. Word2vec is a group of shallow, two-layer neural networks that are trained to represent linguistic and semantic contexts of words. [4] By training the word2vec model on the "corpus" (a structured set of text used for statistical analysis) of tweets, a word is then represented by a corresponding vector that can compare its similarity

of its linguistic meaning to another word (i.e. the word "aren't" is similar to "weren't," "ain't," and "shouldn't").

A common tactic for semantic analysis is to take the average of the word2vec model representations of the words in the sentence to train the convolutional neural network with. Again, this does not best represent the full meaning of the sentence. Instead, the TF-IDF (term frequency-inverse document frequency) statistic is used to calculate a weighted average whose weights depend on the word's relevance to the rest of the model's corpus. [5]

The convolutional neural network itself is a two-layer network. The first layer contains 32 neurons to match an input batch size of 32. The second layer only contains 1 neuron to ensure that there is only one output (as this is a binary classification situation). A convergence study will be done on the epoch iterations in order to determine how many should occur (too little and the model will underfit -- however, too many and the model will overfit).

Results

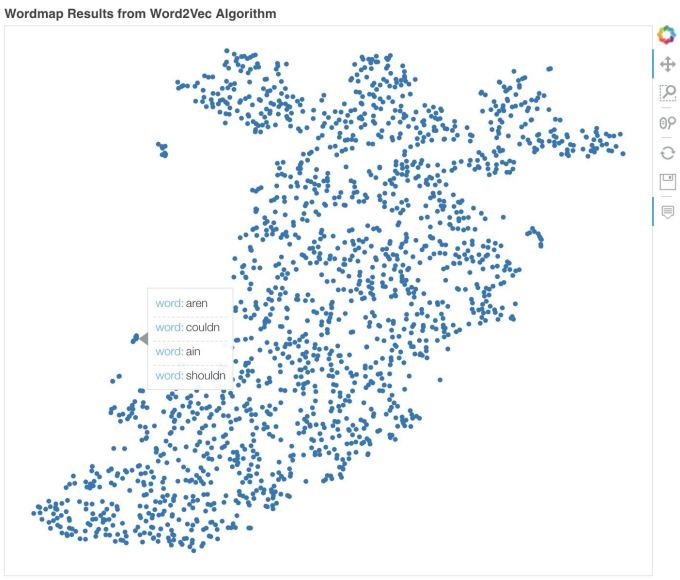

Figure 1: 2-D vector map showing the association of words to other words in the corpus.

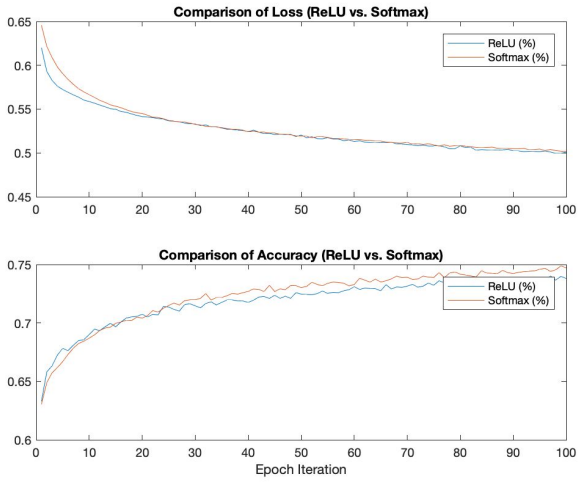

Figure 2: Accuracy and loss percentages calculated every epoch with a 1:5 test-train ratio.

```
************************
* Neural Network Results:
************************
[[1800  577]
 [ 583 1041]]
             precision    recall  f1-score   support

          0       0.76      0.76      0.76      2377
          1       0.64      0.64      0.64      1624

avg / total       0.71      0.71      0.71      4001
```

Figure 3: Confusion matrix and classification report of a 2-layer convolutional neural network with a softmax-driven input layer. The batch size is 32, and the number of epochs is 100. By observing the confusion matrix, it is seen that the model is more likely to accurately label cyberbullies (0.71) than mislabel innocent tweets (0.64).

Discussion

Analysis of Results

The visualization of the vector assignments to each word in the model's corpus can be shown in Figure 1. Words that are similar to each other semantically are represented as being close to each other in space in the word map. For example, the words "aren't", "couldn't", "ain't", and "shouldn't" are all words that, according to

the word2vec model, are similar. Visually confirming this model is important to understanding not only if the vectorization of the words are inputting into the word2vec model properly, but if the associations are accurate. It is possible to deduce similarly identified words through this model, and it is shown that words that are misspelled still have a similarity to their actual, properly-spelled counterpart. This is extremely important for the purpose of this model. If the words were not vectorized like this, it could be inferred that a misspelled word would not be associated with its properly spelled counterpart as readily.

To determine what activation function to properly use for the first neuron layer, a neural network was trained with both rectified linear units (ReLU) and softmax activation functions as their first layer. The results for the first 100 epochs are shown in Figure 2. Initially, it is shown that the softmax activation function-driven model has a greater loss when compared to the ReLU-driven model, while having comparable accuracy values. However, it becomes apparent at roughly 25 epochs that the loss percentage of both models converge to the same value, while the accuracy of the softmax-driven model surpasses that of the ReLU-driven model. Therefore, the neural network is then on to be modeled as a softmax-driven model.

Choosing the epoch amount is an artform in and of itself -- too small of an amount, and the model will be under fitted. Too large, and the model will be overfitted. The number of epochs chosen for this neural network is 100. The confusion matrix and classification report of this neural network is shown in Figure 3.

Moving Forward / Potential Improvements

In Figure 3, it is seen that the model is more likely to accurately label cyberbullies (0.71) than mislabel innocent tweets (0.64). In the real world, it would be best for the classifier to minimize false positives, even at the risk of decreasing the overall accuracy of the model. The activation thresholds could be modified in order to better fit the model's needs.

Furthermore, there are still reservations regarding if the convolutional neural network is set to its maximum potential settings. Implementation of the hyperopt library could allow optimization of every setting of the convolutional neural network, therefore allowing for maximum efficiency. [6]

Although convolutional neural networks are best known for their convolutional ability and expertise at classifying images. Because the matrix construction of word2vec is similar in nature to the matrices used to represent image data, some more processing could have been done on the model before training to smooth out some relations (such as max pooling).

The convolutional neural network is only composed of two layers. While most models can be sufficiently represented with only two layers, the addition of an intermediary layer may improve the expressibility of the model. However, that would require more data to ensure no underfitting.

Lastly, each tweet in the dataset is hand-labeled, with a total summation of 20001 tweets. More data in this dataset can be readily acquired and could greatly improve the reliability and accuracy of the model.

References

[1] Dataturks. 2018. Tweets Dataset for Detection of Cyber-Trolls. (July 2018). https://www.kaggle.com/dataturks/dataset-for-detection-of-cybertrolls

[2] Cournapeau, David. scikit-learn. https://scikit-learn.org/stable/index.html

[3] Chollet, François. Keras: The Python Deep Learning library. https://keras.io/

[4] Goldberg, Yoav; Levy, Omer. "word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method". arXiv:1402.3722.

[5] Seki, Yohei. "Sentence Extraction by tf/idf and Position Weighting from Newspaper Articles" (PDF). National Institute of Informatics.

[6] Anon. Hyperopt. https://hyperopt.github.io/hyperopt/