Robert Werthman
Username: rowe7280
CSCI 5622
Feature Engineering

Initial Observations

The baseline feature extractor tokenizes a sentence and creates a matrix where each row corresponds to a sentence and each column corresponds to the number of words in the sentence. The number of columns is equal to the number of words in the vocabulary of all the sentences. Each sentence then becomes a vector of the number of the number of times each word shows up from the vocabulary.

How I Validated My Changes and My Highest Score

I used 5-folds cross validation of the training set as my validation set and the mean of the 5-folds cross validation score to gauge the accuracy of the changes I made. Although, an increase in the accuracy of the validation set did not always indicate an increase of accuracy of the test set on kaggle. The highest score I achieved was a 5% increase of the baseline of 0.63 to .67.

Things That Improvemed My Score

It looked like I could clean up the text by removing the numbers, punctation, and special characters from each trope. I thought that this would decrease the size of the feature vector and get rid of features like "." or "1" that didn't indicate that a trope was a spoiler or not. I also added stemming to further decrease the size of the feature vector by stemming each of the words. The vocabularly would be made up of stems. For example, instead of "understanding" and "understand" being two separate words in the vocabulary there would only be "understand." I thought this would help narrow down words that provided a good indication whether something was a spoiler or not. I also used lemmatization for that same affect.

What I think made the greatest difference in making the classifier accuracy better was the use of n-grams. N-grams allowed me to gather more context about the trope by including more words that could be compared against test examples. At the same time N-grams increased the number of features that were extracted from each sentence. For example if we had the sentence "Bi-grams are cool!" with n = 2 instead of the vector ["bi", "grams", "are" "cool"] we would have the vector ["bi", "grams", "are", "cool", "bi grams", "grams are", "are cool"].

Things That Did No Really Improve My Score

I did try to use part of speech tagging. I tokenized the sentence into parts of speech and then each feature vector became a count of the parts of speech in the sentence. It looked to me like tropes that were spoilers contained more nouns. But this did not work out and part of speech tagging with part of speech counts only produced an accuracy of just over 50%. Including the word and part of speach as each feature created an accuracy of over 60%. Doing this and including an N-gram model produced about the same.

I also tried to use term frequency-inverse document frequency to evaluate how imporant a word was to a document and weight words that occured less frequently more than common words like "the" or "I." Unfortunately this did not improve the score very much when submitted to kaggle, but the score improved significantly for the testing I did on my local machine though. I also looked at the number of words in a sentence that was a spoiler vs. the number of words in a sentence that was not a spoilers. A spoiler had more words than a sentence that was not a spoiler but using the length of a sentence as a feature only yielded an accuracy just over 50