Using NLP Networks to Detect Satirical News Headlines

By JR Cabansag

OVERVIEW

In this project, we sought to create a network that could detect satirical news headlines, to help prevent human errors of taking joke headlines as fact. In this project, we explored various sentiment analysis NLP networks such as Naive Bayes, Logistic Regression, and Support Vector Machines in order to try to detect satire in headlines. We also explored various feature extraction techniques like Bag -of-Words, Word Counts, and TF-IDF to see which one was most effective. We found that a simple unigram Naive Bayes approach worked best out of all of the models, likely because the dataset was fairly small, and the other, more complicated networks required more training examples to outshine Naive Bayes.

BACKGROUND

There are several news outlets which create news headlines for the sake of comedy - which are not to be taken seriously. However, for various reasons, these news headlines could be taken as fact by some portion of the population. This phenomenon has been widespread enough that it's even earned it's own subreddit, /r/AteTheOnion, which is dedicated to making fun of people who fall for satirical headlines. We wanted to explore this problem and see whether we could prevent the spread of misinformation by creating a network that could try to detect satirical news headlines.

Figure 1. Person unknowingly takes satirical TheOnion headline as fact.

DATASET

We are using a dataset from Kaggle of a series of ~28,600 headlines, labeled as satirical or not. It has ~13,600 satirical headlines all pulled from *TheOnion*, and ~15,000 non-satirical headlines pulled from *HuffPost*.

We split the data into ~20,000 headlines for training, and ~8,600 headlines for testing.

METHODS

We're using various NLP networks, commonly used for sentiment analysis, and analyzing their performance:

NLP Networks

Naive Bayes - a probability classifier which calculates probabilities for headlines by taking tokenized words, and calculating the probabilities of headlines being satirical by basing it off of probabilities in the training set

Logistic Regression - utilizing and optimizing weights for features in the training set to achieve the absolute minimum loss on the training set

Support Vector Machines - looking at the training set and finding a plane partition that best separates the labeled data into its associated parts

We also looked at various feature extraction methods to see whether they affected performance when used to create different feature vectors for Logistic Regression and Support Vector Machines:

Feature Extraction

Bag Of Words - a vector of size |vocabulary| marking 1 for all tokenized words present in the headline, 0 otherwise

Word Count - a vector of size |vocabulary| marking the number of occurences of tokenized words in a headline for words that are present in the headline

TF IDF - a vector of size |vocabulary| marking a weighting for words, calculated by the log of the frequency of the word in the headline, multiplied by the log inverse of the number of documents the word has been present in.
*(log(# of word occurrence in headline + 1))**
(log((1+# of total documents)/(1+# of documents word occurs in))+1)

REFERENCES

モルラ, モハマド Meraj. "Machine Learning: Sentiment Analysis of Movie Reviews Using LogisticRegression." Medium, ITNEXT, 12 Oct. 2018, https://itnext.io/machine-learning-sentiment-analysis-of-movie-reviews-using-logisticregression-62e9622b4532.

Misra, Rishabh, and Arora Prahal. "Sarcasm Detection Using Hybrid Neural Network." Kaggle, 2019, https://www.kaggle.com/rmisra/news-headlines-dataset-for-sarcasm-detection.

Reddy, Vasista. "Sentiment Analysis Using SVM." Medium, Medium, 25 Nov. 2019, https://medium.com/@vasista/sentiment-analysis-using-svm-338d418e3ff1.

Stecanella, Bruno. "A Practical Explanation of a Naive Bayes Classifier." MonkeyLearn Blog, 20 May 2019, https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/.

RESULTS

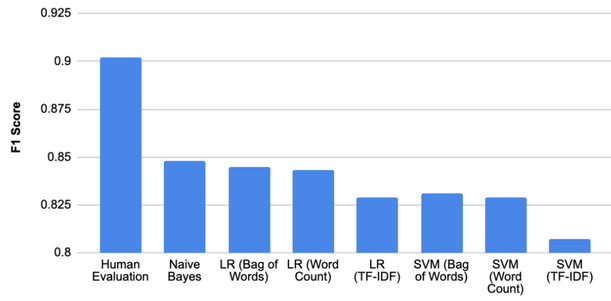

Figure 2. F1 scores of different NLP networks and feature extractions.

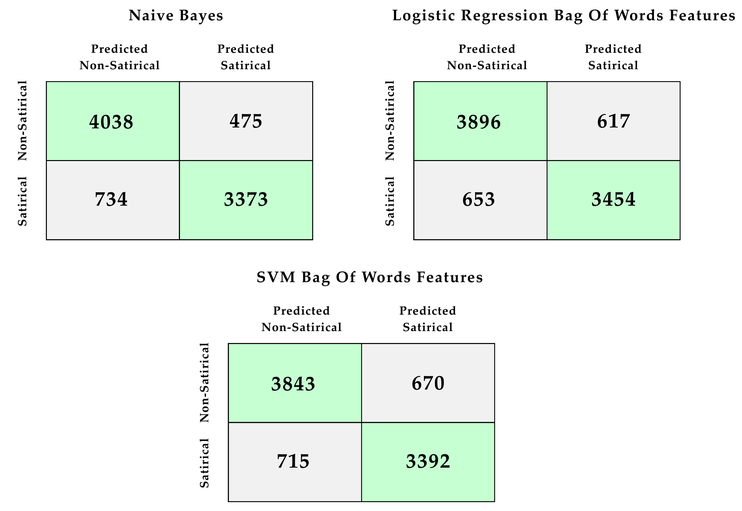

Figure 3. Confusion Matrices of NLP networks with their best performing feature extraction methods.

Out of all of our networks, our most simple one, the Naive Bayes classifier, performed the best according to our F1 score metric. We believe that this was because the training set was relatively small and simplistiic for the more complicated networks to fully shine. We believe that in the future, if we collect headlines from a wider variety of news sources, that logistic regression and SVM could possibly meet or even exceed Naive Bayes' performance. In the meantime however, our results show that more complicated NLP networks are not always better.