Finding Commonalities in Misinformative Articles Across Topics

1. Introduction to the dataset:

Our data is collected from http://fakenews.research.sfu.ca/#parseWebs where we use the datasets containing Snopes, Politifact, and Emergent.info articles of varying real and fake news from 2010 to 2018. We took articles from each dataset to create a new dataset that contains real and fake news for specific genres of news. We gathered news about 100 data for each political and scientific topic from the Snopes, Politifact, and Emergent.info datasets to use as our training dataset. We also created a dataset filled with varying topics to use as our testing dataset. Our plan with these datasets is to find commonalities of misinformation across different topics. To do this, we are training our models based on set genres and then testing the results on a set of data with varying genres of news.

2. Identify predictive tasks:

For our research, we are using our training dataset to predict whether a random article, regardless of the genre, is misinformative or not. We will train our models so that it learns the commonalities of misinformation for a set topic. Then we will test our findings onto a random article to see if our model can accurately predict whether that article is misinformative or not. We use the scores of the Decision Tree, Logistic Regression, Random Forest Classifier, and SVM to test our models' accuracies. In addition to examining accuracies, we will look at the intersection of a list of words that each model deems most important to determine if an article is misinformative, this will help figure out which topics have common indicators of misinformation. After our models make a prediction on a random genre article, we want to examine differences of misinformation across different genres of news.

3. Describe your model

The following models are the same models we used in our replication project in the previous quarter. We use these models since we are familiar with them. Some models have been removed, that we feel like are not as useful for our task.

The Decision Tree model utilizes the structure of a tree to classify data. It has branches and leaves which are the classified data path. The Decision Tree model makes a prediction based on the learnings of the decision rules from resulting features of data. We use sklearn for our

Decision Tree classifier since it has the option to set the max depth. Having this option allows us to shorten the time for processing this model.

Binary Logistic Regression utilizes linear regression function which is modified to scale any data a value in between 0 and 1. The value assigned is the probability of the prediction belonging to class 1 or 0. We use sklearn implementation of Logistic Regression since linear regression is regularized to prevent overfitting.

The Random Forest Classifier is an estimator. The classifier fits multiple decision trees on smaller sub-samples of the dataset to get a different approach compared to a regular decision tree. Additionally, the Random Forest Classifier averages result to control overfitting and improve the accuracy of predictions.

A Support Vector Machine (SVM) searches a hyperplane in N-dimensional space to classify particular data points from a dataset. The SVM has updatable gradients for the weights when classifying data points. We use sklearn's SVM since it is regularized to prevent overfitting.

4. Literature

For some parts of our project, we relied on some formatting and testing of a covid misinformation report by Sajad Dadgar, titled A COVID-19 misinformation detection system on twitter using network & content mining perspective (https://github.com/sajaddadgar/A-COVID-19-misinformation-detection-system-on-Twitter-using-network-content-mining-perspective) . We utilized some preprocessing utilities as well as what models to focus on for our news misinformation detection. In addition, we implemented his grid search method for finding optimal parameters for the appropriate models. Finally, we looked at his testing methods and how to display results for finding what model to use. While these were first geared towards twitter posts with mainly lower amounts of text/characters for analysis, we found that it could help with denser articles that contain more details on the subject at hand.

5. Results and conclusions

When examining the accuracies of the models, we decided to show both ends of performance for each topic, the best and the worst. The best performing model for the general classifier was SVM with an APR[1] score of: 61.4%, 76.5%, 35.1% respectively. The worst model was Random Forest with an APR score of 50.7%, 52%, and 35.1%. The best performing model

[1] APR stands for Accuracy Precision and Recall, these are the scores of which the model is evaluated by in the project

for Science was Decision Tree with an APR score of 73.9%, 69.2%, and 81.8%. The worst performing model was Logisitic Regression with an APR score of 56.5%, 1.0, 9.1%. For politics our best model was Random Forest with an APR score of 60%, 83.3%, 35.7%. The worst model was Decision Tree with an APR score of 52%, 55.5%, 71.4%. For economics our best model was Decision tree with an APR score of 60%, 50%, 50%. The worst model was Random Forest with an APR score of 48%, 42%, 80%.

Works Cited

Dadgar, Sajad.
"Sajaddadgar/A-Covid-19-Misinformation-Detection-System-on-Twitter-Using-Network-Content-Mining-Perspective." *GitHub*, https://github.com/sajaddadgar/A-COVID-19-misinformation-detection-system-on-Twitter-using-network-content-mining-perspective.