The Divide Between Falsity and Honesty
A Classifier for Fake News

Outline:

We hope to create a classifier for articles, with the intention of categorizing such documents as either fake news (propaganda, misinformation, etc) or the truth. Our core research question is, "what features and characteristics of a piece of text showcase dishonesty or malintent?" Understanding the key differences between something that is purposely-biased information (the mentioning of political figures, politically charged words, etc) and something that is meant to honor the truth can allow us to computationally see through facades and discover what is meant to propagate negative reaction and wrongful spread of falsities. As for our data, we have news articles from unreliable sources (fake news), and news articles from Reuters as a more neutral, unbiased source. Our dataset contains 23,481 fake news articles and 21,417 'true' articles. For the fake news, we have an average document length (in terms of the actual article content, 'text') of 2547.4 tokens, with 74.3% of those documents being distinct, and 53.3% being unique. As for true news, we have an average length of 2383.3 tokens, with 98.9% of those documents being distinct, and 98% being unique.. The 'distinction' in question is the percentage of values that are different from each other, where the 'uniqueness' is the percentage of those that have no duplicates. We believe the lower percentages for fake news has much to do with the culture of misinformation being spread: lots of biased news sources appear to source their information from other, similarly-biased sources; this can lead to very similar, if not identical, titles and texts. With our text metadata label explained, we also plan to use 'title,' which has mean length of 64.7 tokens and a range of 107 tokens, as well as a distinction of 97.2% and a uniqueness of 94.8% for true, and 94.2 tokens for fake with a much wider range of 278, along with distinction of 76.2% and uniqueness of 54.5%. 'Subject' is one of 5 possible subjects, with fake news having 'News' being the most common; 'politics' follows right behind, with, interestingly enough, 'left-news' coming soon after. True news only has two subjects: 'politicsNews' and 'worldnews,' both of which are very similar in distribution. Finally, the 'date' label is quite straightforward; our true news ranges from 2016 to 2017 news articles, while fake news ranges from 2015 to 2017, its biggest frequency being in 2016. All of this information was gathered using the PandasDB library to explore and understand our data. In addition, we are utilizing Git as a version history tracker, to monitor our progress and make coordinating tasks easier, with less problems and merge conflicts. To transform our documents, we are using the Natural Language Toolkit (NLTK) library to tokenize our words, and Numpy to vectorize our texts.. We plan to curate our vocabulary list to exclude stop words, incredibly common

words, and rarer words. We also plan to explore the sentiment feature of our text corpus to see how sentiment is distributed between the two sides of our data. As previously explained, our metadata labels have a lot of aspects to analyze as well; in terms of feature analysis, we plan to see how our vocabularies and sentiments correlate with the metadata so as to confirm their purposefulness and connection. This will include evaluating our text vectors as features as well, using a Naive Bayes classifier. In order to build our classifier, we plan on utilizing subsets of the true and fake data to train, and then use it to classify the remaining articles. We hope to experiment with certain concepts learned within the course for clustering. To start, we plan to visualize our data using Truncated SCV and t-SNE, before applying agglomerative and k-means clustering, and plotting using the hierarchical clustering dendrogram. This will help us better visualize the frequencies of certain features. Using this, we can train our classifier using a subset of our articles, and attempt to use it on the rest of the set. We have a lot planned to truly do everything we possibly can with this dataset, and build a great classifier.

Status:

Our first steps towards success were setting up a repository on Github for the two of us to utilize in order to better keep track of both our progress, individual contributions, and avoid any types of merge conflicts. Simply put, trying to do this project without a repository would've proven ill-planned. Next, we created a scratch jupyter notebook to use to understand and figure out our data; this scratch will likely become what we do our work in. However, we may end up splitting parts of our work into other notebooks. As for now, we imported multiple libraries, including REGEX, Numpy, Pandas, Scipy, NLTK, SKLearn, etc. We then read in our two data files (true, false) as Pandas dataframes, which allowed us to utilize a function called ProfileReport, which generates an HTML with plenty of useful information about our metadata, the contents and contexts of each label, and the dataframe as a whole. With this, we have gone ahead and begun building corpuses of the two sets. Thus far, we have tokenized both by eliminating all stop words and punctuations, and are now taking a look at common words. After this, we will vectorize our features and analyze sentiments within the sets. This will lead into creating our Naive Bayes classifier to check the quality of our vectors. Next, we will begin visualizing our vectors before building cluster necessities to then implement both k-means and agglomerative clustering techniques to better understand and classify our data. Finally, we can build our classifier using a subset of our articles, and try to classify the rest of the set, and take a look at our margin of error. This will, in the end, show us what exactly can be a determinant of an article being mal-intensive vs. constructive. This will also lead into us using any extra free time to do last minute experiments on our data.