Classifying News Articles Using Machine Learning

Professor Alfeld, Annabelle Gary '20, Jason Greenfield '20, Samantha Rydzewski '21, Lesley Zheng '21

Introduction

In response to our nation's media crisis, we attempted to use machine learning to quantify and classify news articles. We initially examined a dataset of real and fake news from MIT Lincoln Labs, but then shifted our focus to classifying articles by source.

What is Machine Learning?

Machine learning (ML) is a tool which takes real world experiences, data, and past observations and uses them to predict an outcome. For example, fake news detectors take in labeled news articles (real or fake) and use their characteristics (e.g., image forensics, source, content) to predict whether a new, unseen article is fake. We used ML to predict from what source thousands of news articles came from.

Data Collection

We used Selenium Webdriver to automatically crawl through news sites. Then, we visited each article and scraped the desired data.

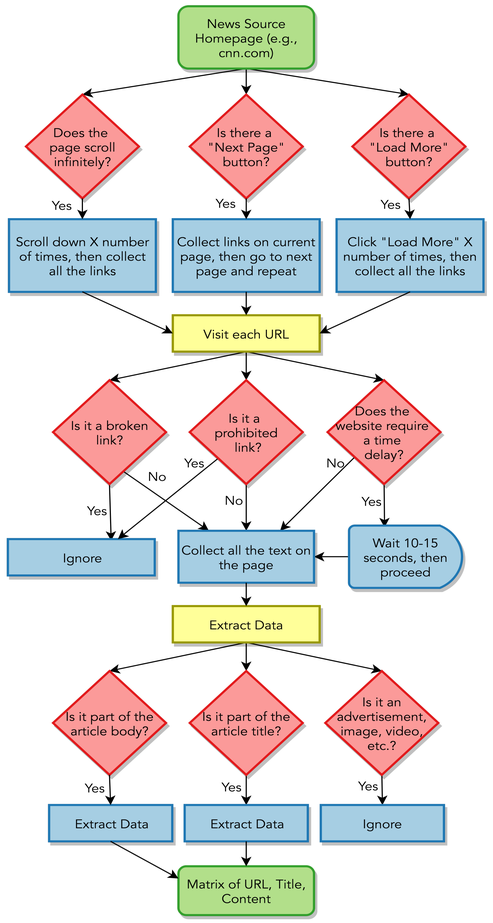

Figure 1: A diagram that shows the process of how our bots collected the data.

Data Processing

We created a list of 52 handcrafted features (number of quotes, adjectives, places, etc.) indicative of writing style and another list identifying the 10,000 most common words across the entire dataset. We then calculated the values of the 52 features and the frequency of the 10,000 words for each article to construct two separate data sets.

Key:
Geographic Locations
Numbers
Special Characters
Punctuation
Noun
Adjective
Verb

In Pennsylvania, as well as in most of Maryland, and parts of the Virginia and North Carolina coastlines, there is a moderate chance (20-50%) of rainfall exceeding flash flood guidance.

Figure 2: A representation of data processing on a sentence from a CNN article (https://www.cnn.com/2018/07/24/weather/mid-atlantic-flooding-wxc/index.html).

Dimensionality Reduction

We visualized our data by reducing the 52 and 10,000 features to two or three dimensions using the dimensionality reduction techniques Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), and Isomap.

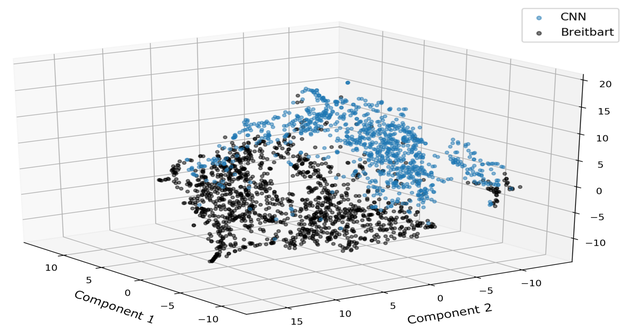

Figure 3: We used PCA to compute the ten principal components (the linear combinations of features that capture the most variance) from the 52-feature matrix of CNN and Breitbart articles. Then, we applied t-SNE to further reduce the matrix to three dimensions. Finally, we plotted the data points and colored them based on source.

Learning Models

1. Our goal was to teach the computer how to classify news articles by source.
2. To do so, we trained the computer on two thirds of our data with three different learning models.
3. We then tested the models on the remaining third and saw varying results

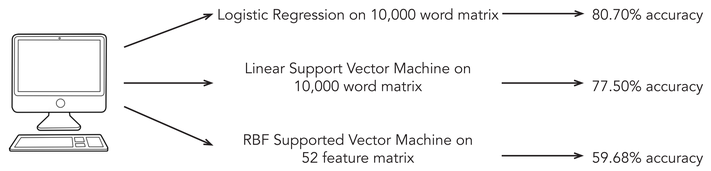

Figure 4: A summary of how we used learning models in our research.

Results and Conclusion

Our plots showed significant clustering and our learning models were impressively accurate given the complexity of this eight-class classification task. However, our results were partially undermined by non-thoroughly cleaned data. For example, some of the scraped articles included special quotation marks or source names. This allowed the computer to classify articles by these two features alone. We were able to correct for most of these effects but not all of them.

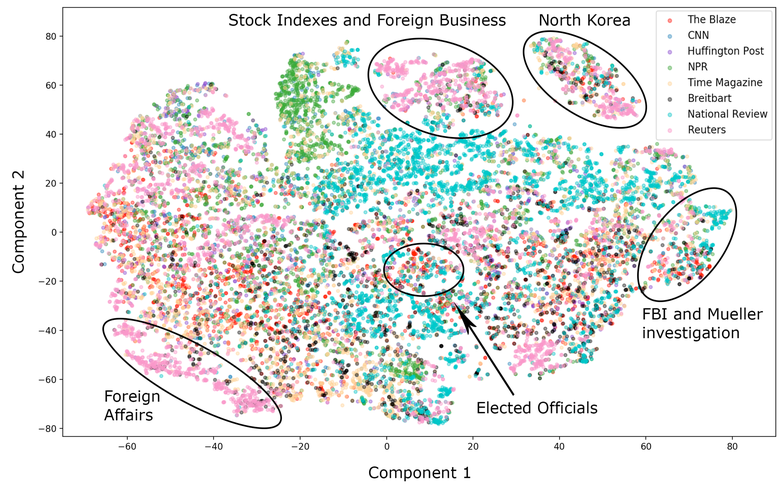

Figure 5: We used PCA to compute ten principal components from the 10,000-word matrix with articles from all sources. Then, we applied t-SNE to further reduce the matrix to two dimensions. We plotted the data points and identified clusters of articles with similar topics.

We used a Random Forest Classifier to calculate, for the 10,000-word matrix, the most influential words. The top two words, "editing" and "reporting," had a disproportionate influence on the models because every Reuters article names its reporters and editors. For the 52-feature matrix, the features with the greatest weights were: 1) average number of words in a sentence, 2) number of sentences in an article, and 3) average number of letters in each word.

Future Directions

We would like to: 1) include more sources and features, 2) account for each feature's influence, 3) produce cleaner data, 4) analyze political sentiment (i.e., left-leaning vs. right-leaning).

Acknowledgments

Thank you to Professor Alfeld, Amherst College Summer Undergraduate Research Fellowship, MIT Lincoln Labs, and the creators of the following libraries: Scikit-Learn, Selenium, Matplotlib, Natural Language Toolkit, Pickle, and NumPy.