Fake News Project Report

Introduction

Problem Statement

In order to stop the spread of fake news, can social media websites use an algorithm that outperforms human fake news detection by using NLP to determine whether an article is real or fake?

Background

With the rise of social media, fake news has run rampant online due to the ease of sharing articles to a massive amount of people with the click of a button. The spread of fake news has become a major concern with 67% of Americans identifying social media as their main news source (1). Because fake news is created with the intention of deception, it can be hard to tell whether an article is real or fake. In fact, studies show that humans are not very good at detecting fake news, reporting accuracy rates of 55-58% (1), which is just above chance. Without researching the information or the source, the average person cannot tell the difference between what is false and what is true; thus many people and websites spread fake news unintentionally.

Fake news can have major negative effects including political influence, fear mongering, and compromised public safety. For example, fake news may have had a critical role in the 2016 presidential election campaign. For an idea of the reach of false news during the election campaign, the top 20 false news stories garnered 1.4 million more engagements (i.e. likes, comments, and shares) than the top 20 election news stories posted by major news sites (2). In addition, some fake news is written to stir up fear and chaos. In 2017, Hurricane Irma was believed to be a 'category six' hurricane from a fake news article shared more than 2 million times, even though there are only five possible categories of hurricanes (3). On the other side of the spectrum, fake news about COVID-19 led people to believe that the virus was a hoax (4), a false notion with disastrous consequences as Americans did not take the virus seriously enough to stop the virus from spreading and killing hundreds of thousands of Americans. The proliferation of fake news online has had very real consequences, contributing to mass confusion and distortion of reality.

Goal

It is clear that the effects of fake news are damaging, and the average person cannot rely on their own judgement to assess the validity of an article; therefore, a website or company's ability to detect fake articles is extremely important in order to curtail the spread of fake news. Social media websites need to be held accountable for spreading fake news, and with the help of machine learning, websites may be able to automatically filter out fake news before it spreads. Below, I will outline a method for fake news detection that can be adapted by social media websites in an effort to contain the fake news epidemic.

Approach

Data Wrangling

For this project, I used preexisting data created by the University of Victoria (5). The database contained one dataset of 21,417 'true' articles collected from Reuters during 2016 and 2017. Reuters is a highly reputable, non-biased news source, and therefore, very unlikely to publish fake news. There was a second dataset of 23,481 'fake' articles compiled from various 'news' sources that were flagged as unreliable by fact-checking websites including Wikipedia and Politifact. The datasets can be downloaded from the University of Victoria ISOT Lab website as an excel file.

The data contained 4 columns for date published, subject, text, and title. Upon first exploring each dataset, I noticed that the Reuters articles had a specific format; namely, most true articles had "[SATE] (Reuters) –" at the beginning of every article, so I removed this phrase with some text processing so that the algorithm did not pick up on Reuters-specific formatting. Then I created a label column for each data set, declaring if the article was true or fake, and then I combined the datasets. After combining the datasets, I dropped the 'subject' column because it did not contain any relevant information. I also dropped 5,793 rows of duplicates along with 1 row of that did not have text, ending up with 39,104 rows of data.

Feature Engineering

I used NLP to extract features from the text, title, and date columns. For the basic text features, I extracted the character counts, word counts (including the total number of words, stop words, and uppercase words), average word length, punctuation counts (!, @, ?, .), and numeric counts for each text and title separately. I also extracted the day of week published and month published from the date column. Year published would not be useful for fake news detection. After extracting these basic features, I explored the data.

Exploratory Data Analysis

- The data was relatively evenly distributed with 54% of the data labeled as real and 46% of the data is labeled as fake

- Character counts, word counts, period counts, and stop word counts were all correlated, as shown in figure 1 below, so I dropped character counts and period counts and transformed stop word counts into stop word percentage in order to decorrelate stop words and word counts

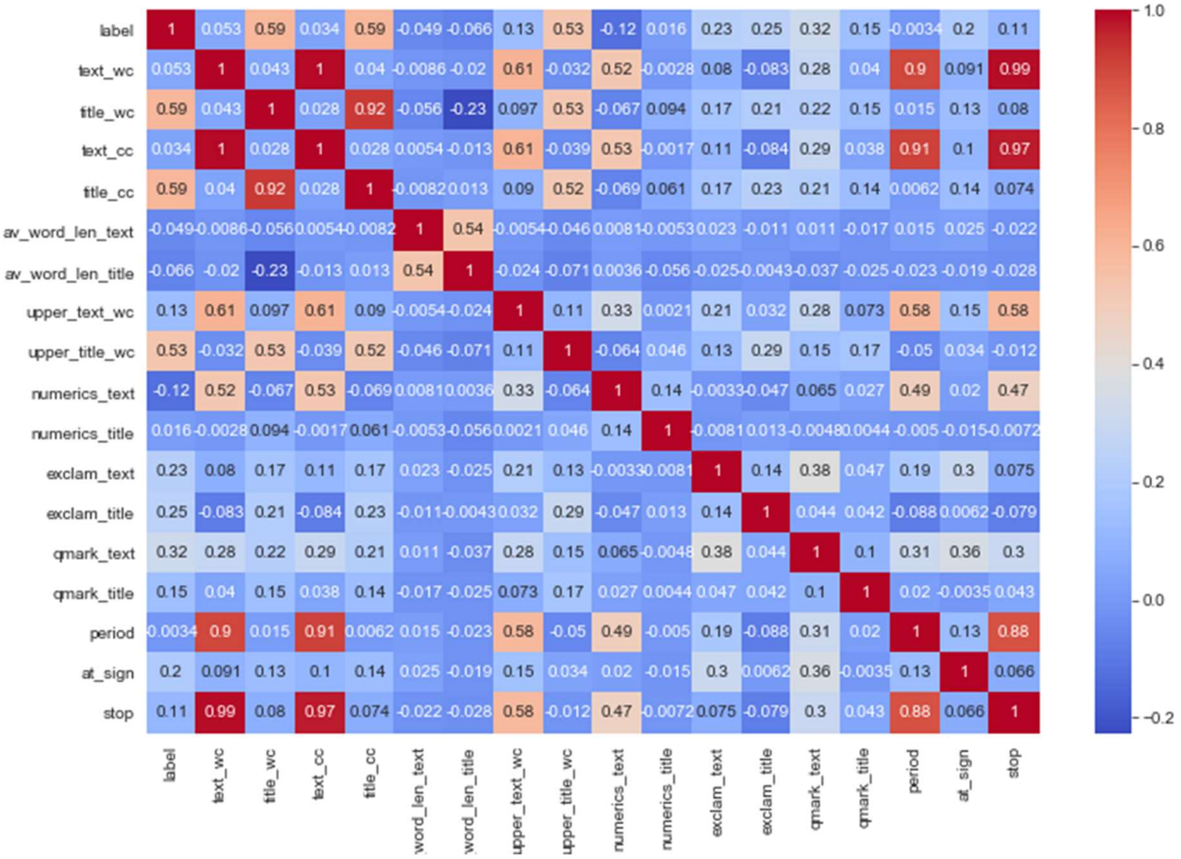


Figure 1: Heatmap of feature correlations. Warmer colors signify higher positive correlations.

- Word count distribution plot showed that real news articles had shorter titles than fake news articles

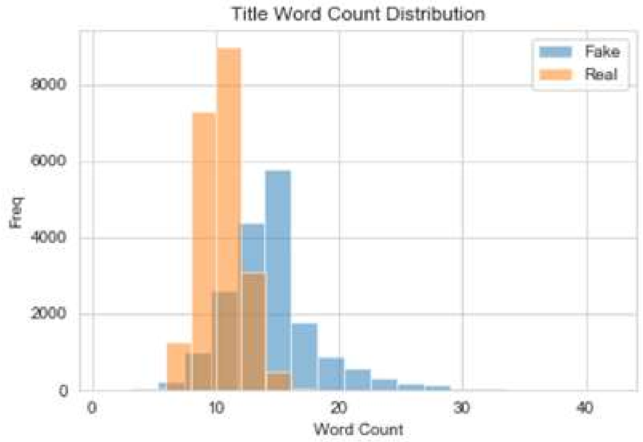


Figure 2: real news title lengths vs. fake news title lengths

- Counts of uppercase words signified that fake news contains more uppercase words

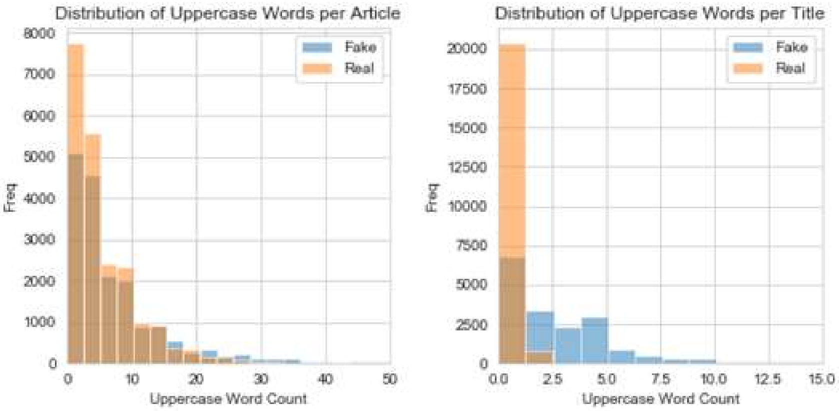


Figure 3: count of uppercase words in real articles vs. fake articles

- There are more stop words in fake articles

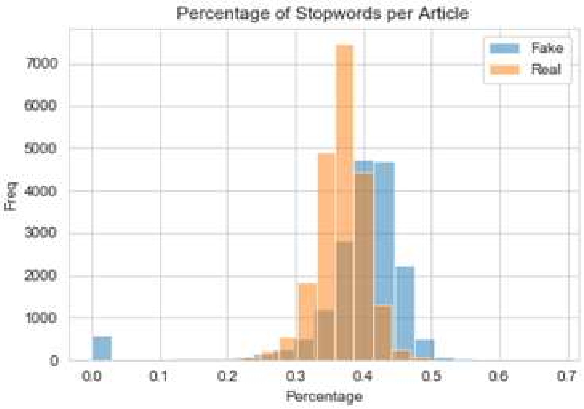


Figure 4: percentage of stop words in fake vs. real news articles

- There is more punctuation usage in fake articles

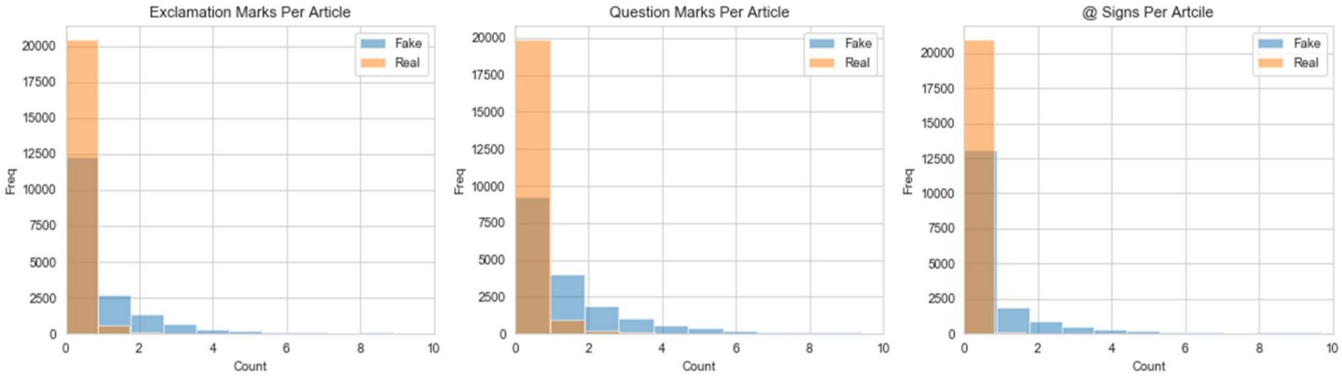


Figure 5: punctuation counts in real articles vs. fake articles

- Fake news is more likely to be published on a weekend than real news, and there is more fake news published at the beginning of the year than real news.

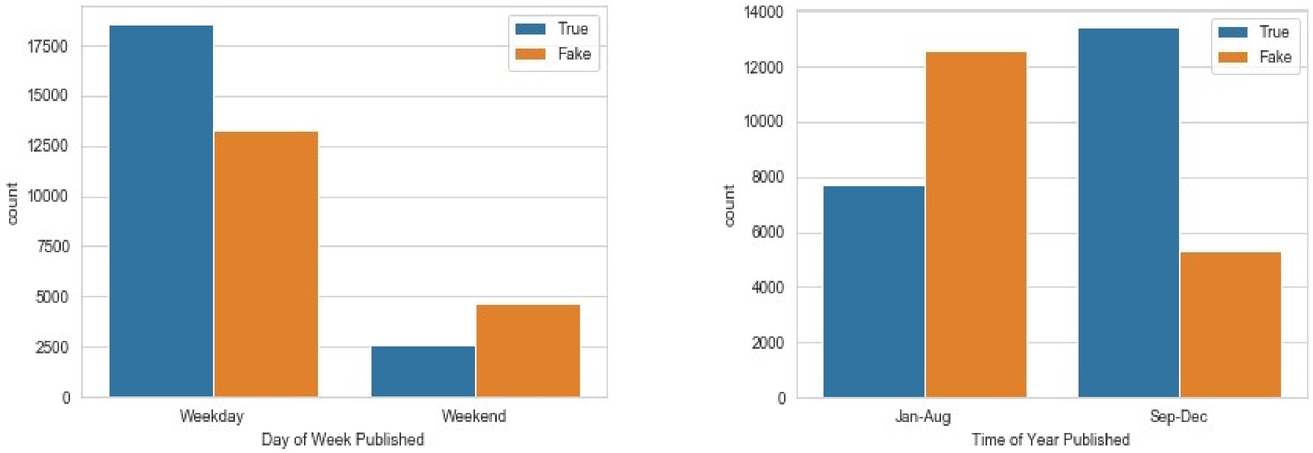


Figure 6: Publishing times for real articles vs. fake articles

- Non parametric hypothesis testing confirmed preliminary conclusions of exploratory analysis, and all findings were determined to be statistically significant (p< .001), informing that these features should be used in the model.

Pre-processing and Additional Analysis

For text processing, I made all words lowercase, removed punctuation, numbers, and stop words, and I lemmatized all rows for the text and title columns. Then I tokenized every word with sklearn's CountVectorizer. I noticed that two of the top words used were 'reuters' and 'via' which are words very commonly used in Reuters articles but not true for most real news articles, so I removed these words. The most frequently used words in the titles were 'trump', 'video', 'say', 'house', and 'obama.' The most frequently used words in the text were 'said', 'trump', 'state', 'would', and 'president'. I wanted to compare real vs. fake and title vs. text, so I created a word cloud for all 4 combinations with WordCloud, shown below in figure 7.

Figure 7a: word cloud of real news articles

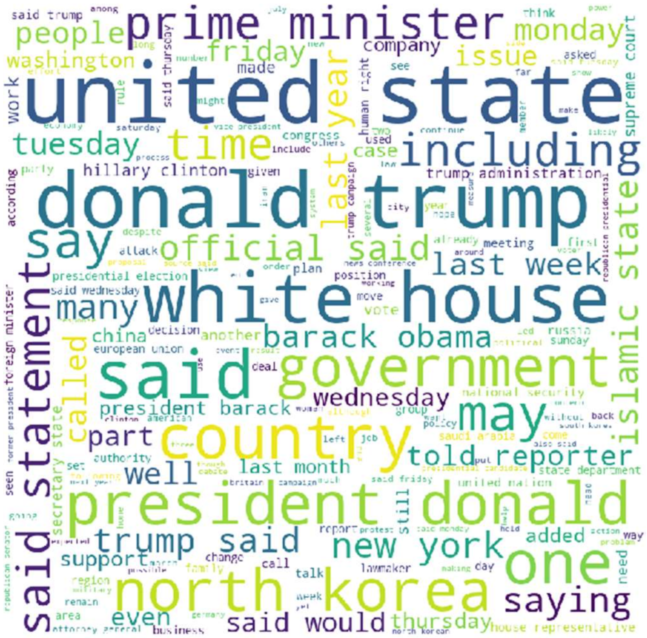
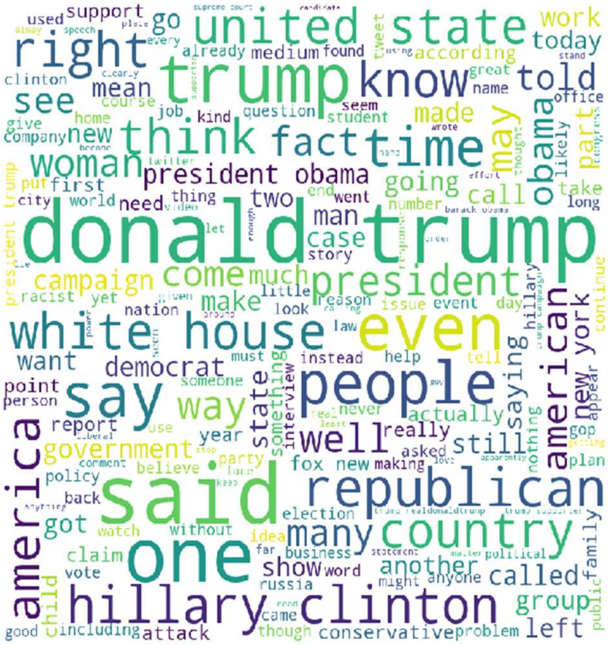


Figure 7b: word cloud of fake news articles



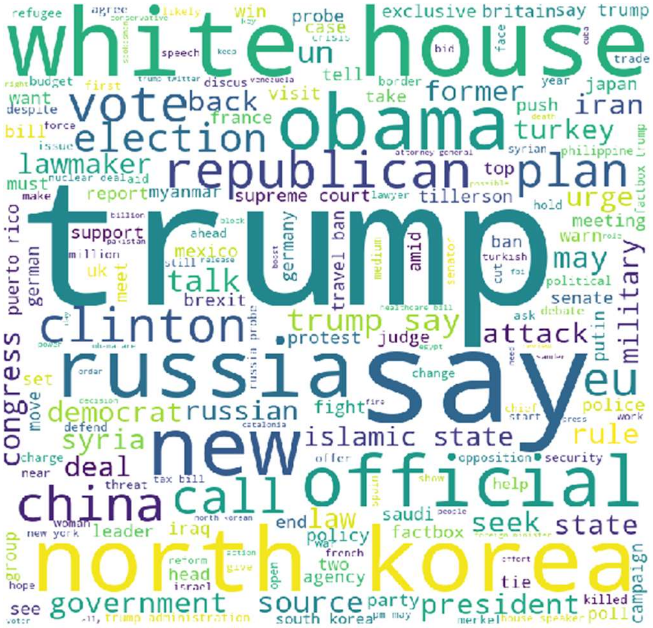


Figure 7c: word cloud of real news titles

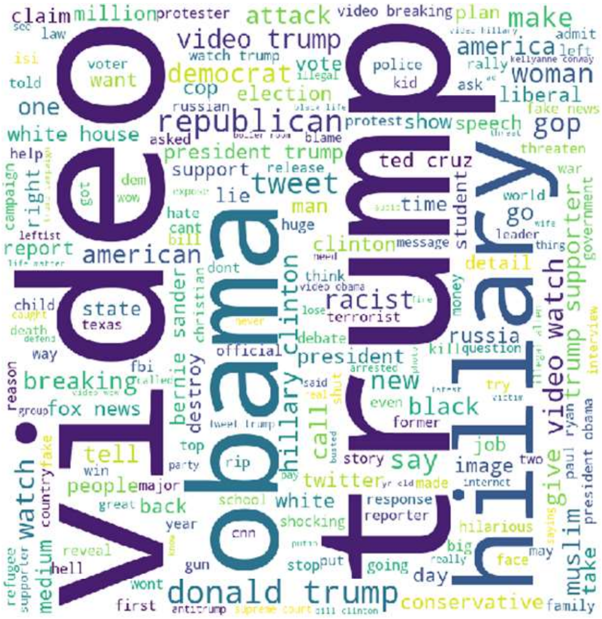


Figure 7d: word cloud of fake news titles

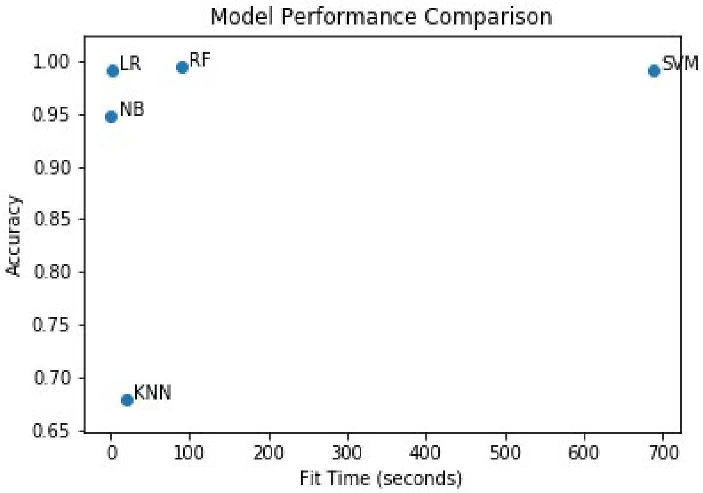
These word clouds are very enlightening. First, there are a lot more ‘trigger’ words used in fake news titles vs. real news titles. For instance, the fake news title word cloud is littered with words such as ‘watch’, ‘video’, ‘hell’, ‘terrorist’, ‘lie’, ‘breaking’, ‘threaten’, ‘busted’, ‘shocking’, and ‘destroy’, while these words are not seen in the real news title word cloud. This is because these are high-emotion ‘clickbait’ words that stir up a reaction and lead people to click on the article and read it, a very common tactic for fake news. In addition, there are more mentions of people in fake news articles (e.g. ‘trump’, ‘hillary’, ‘obama’, ‘ryan paul’, ‘kellyanne conway’, ‘ted cruz’, ‘bernie sanders’), while in real news titles, there are more mentions of places and real world news events (e.g. ‘brexit’, ‘travel ban’, ‘north korea’, ‘germany’, ‘iran’, ‘turkey’, etc.). I postulate this is because it is more believable and more interesting to read fake news about a scandal of a high profile person than it is about a country or international events which can be easily fact-checked and are not as interesting to read. Also, the words used in fake news articles tend to be less formal. For example, ‘hillary’ is a top word in fake news titles while ‘clinton’ is a top word in real news articles. Another interesting observation from the word clouds is that articles mentioning ‘fake news’ are more likely to be fake news. It also appears that ‘fox news’ is more likely to be mentioned in fake news articles. I presume this is because any news that lists Fox News as a source is more likely to be fake news as Fox News creates extremely biased and unreliable information. Finally, and unsurprisingly, Donald Trump was the most talked about across the board. Perhaps this is because he is the President of the United States, but more likely because his scandalous actions alone are enough to attract readers, no clickbait necessary. The word clouds revealed that fake news is more likely to mention ‘clickbait’ words, unreliable news sources, high profile people instead of places or world events, and more likely to have informal word usage.

In addition to word clouds and tokenization, I also performed sentiment analysis on real vs. fake news. I found no notable difference between fake news and real news in this regard, so I did not include sentiment analysis in the final model. I also performed tfidf vectorization with sklearn using unigrams and bigrams, expecting that tfidf vectorization may perform better than count vectorization. I limited the number of text features to 800 and the number of title features to 200 for a total of 1000 tfidf features along with the 16 basic features included in the final model. Lastly, I standardized the data and split 75% of the data into a training set and 25% of the data into a testing set.

Modeling

I tested out a few different models on the data. First I trained a Naïve Bayes model because of its fast performance and reached an accuracy of 94% out of the box. Then I trained a KNN model, which did not perform as well with 68% accuracy. I then tried using SVM, logistic regression, and random forest. These models all performed extremely well, with accuracy above 99%; however, SVM was very computationally expensive and took over 11 minutes to train a single model. Random forest and logistic

Figure 8: comparison of model accuracy vs. computation time



regression were the most plausible algorithms due to their extremely high out-of-the-box performance and fast computation times; thus, I performed a hyperparameter grid search with 5 fold cross validation to tune both models. After tuning, Random forest achieved 99.4% accuracy with a fit time of 1.5 minutes, and logistic regression achieved 99.1% accuracy with a fit time of 1.4 seconds. Although random forest had the highest accuracy, logistic regression was exceptionally faster with comparable accuracy.

Conclusion

There are key differences between fake news and real news. On average, fake articles have higher word counts, more uppercase words, punctuation, and stop words while real articles have longer words and more numbers. In addition, fake articles and real articles have different times that they are more likely to be published, and the verbiage of fake articles and news articles differ, as seen in word clouds. These conclusions are confirmed by impressive classification accuracy rates of over 99% using simple machine learning algorithms. Thus, automatic fake news detection is very plausible with basic text features, tfidf vectorization, and a logistic regression algorithm.

Fake news detection is an important area of research with a few different applications. Because it can be a cumbersome task for users to trace the article back to its origin, it is important for trusted companies and websites that share news articles on social media to vet their sources before sharing articles. Passing sources through a fake news detector could be used to compile a list of websites to avoid using as a source. Additionally, social media websites could use the model to automatically filter out articles that are detected as fake news before it is spread. Another application for social media is that the fake news detector could be used to spread awareness and warn users of fake news as a notification before the user reads the article. This may help convince the user not to share the article.

Areas of future research could focus on extracting other features from the text, analyzing comments, and minimizing features. Identifying certain styles of writing between real and fake articles may help classification. For example, fake news and real news may have differences in sentence structure, or fake news may have more incorrect grammar and spelling. In addition, user comments on articles could be explored as a method to enhance classification. For instance, if there are many users declaring that an article is fake news, it may certainly be fake news. Lastly, one may be able to improve performance by decreasing the number of features gathered from tfidf vectorization, focusing only on the most important words.

Sources

1. Xinyi Zhou and Reza Zafarani. 2018. Fake News: A Survey of Research, Detection Methods, and Opportunities. ACM Comput. Surv. 1, 1 (December 2018), 40 pages.

2. Craig Silverman. 2016. This Analysis Shows How Viral Fake Election News Stories Outperformed Real News On Facebook. (November 2016). Retrieved July 1, 2020 from https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook

3. Maggie Astor. 2017. No, Hurricane Irma Won't Be a 'Category 6' Storm. (September 2017). Retrieved July 1, 2020 from https://www.nytimes.com/2017/09/06/us/hurricane-irma-category-six

4. Chrysalis L. Wright. 2020. COVID-19 Fake News and Its Impact on Consumers. (April 2020). Retrieved July 1, 2020 from https://www.psychologytoday.com/us/blog/everyday-media/202004/covid-19-fake-news-and-its-impact-consumers

5. Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.