SMU Data Science Review

Volume 1 | Number 4 Article 13

2018

Finding Truth in Fake News: Reverse Plagiarism and other Models of Classification

Matthew Przybyla
Southern Methodist University, mprzybyla@smu.edu

David Tran
Southern Methodist University, davidtran@smu.edu

Amber Whelpley
Southern Methodist University, awhelpley@smu.edu

Daniel W. Engels
Southern Methodist University, dwe@smu.edu

Finding Truth in Fake News: Reverse Plagiarism and other Models of Classification

Amber Whelpley, David Tran, Matthew Przybyla, Daniel Engels
Master of Science in Data Science, Southern Methodist University
Dallas, Texas USA
{awhelpley, davidtran, mprzybyla, dwe}@smu.edu

Abstract. As the digital age creates new ways of spreading news, fake stories are propagated to widen audiences. A majority of people obtain both fake and truthful news without knowing which is which. There is not currently a reliable and efficient method to identify "fake news". Several ways of detecting fake news have been produced, but the various algorithms have low accuracy of detection and the definition of what makes a news item 'fake' remains unclear. In this paper, we propose a new method of detecting on of fake news through comparison to other news items on the same topic, as well as performing logistic regression and multinomial naïve Bayes classification. From the techniques and methodologies, we found that fake news can be classified in the simplest terms as fact-based or non-fact-based. Our model, built upon reverse plagiarism and natural language processing, produces positive results but is not as effective as logistic regression and multinomial naïve Bayes. These models classify fake news more correctly and efficiently than a human could and show that fake news is easily identifiable. The traditional classification models outperform the reverse plagiarism method, but improvements and refinements can be made.

1 Introduction

News content is delivered in many forms, such as television, newspaper, and online articles. With the growing usage of online users, online articles have grown in popularity. Users can read content directly from news corporations' websites or even on social media. Facebook and Twitter are two popular social media sources that people rely on as being authentic and accurate, but the platforms are highly susceptible to fake news. Unfortunately, the modern world not only allows people everywhere to see news items they would never have been aware of in the past, but also provides vast opportunities to lie, trick, and otherwise mislead readers.

Fake news is any form of information about current events that is either fabricated or contains bias. In recent years and influential events, the amount of fake news has spread its presence more easily on social media. The readers might unwillingly read fake news articles without their knowledge. Fake news can take forms directly or indirectly.

According to Benedictine University [1], there are four categories of fake news: false news, misleading news, clickbait, and satire. First, false news is an article or other news item that is blatantly false. False news may involve headlines or images intended

to trigger strong emotional reactions in the readers in an attempt to raise profits or simply to spread the false news further throughout social media via likes or shares. Second, misleading news is intended to steer the audience toward a particular opinion or view on an event. This differs from false news in that it is often loosely based upon truthful events, with the facts skewed to portray an untruthful result or outcome. Third, clickbait is the practice of using or creating an enticing headline or photo that over-exaggerates or does not relate to the news article's content in any way. Last, satire is when an article contains information that is written with a comical tone that readers may misinterpret as being factual.

Each of the previously mentioned forms of fake news are all incorrect and non-factual in some way. Whether the reader impulsively clicks on the news articles due to a bogus headline or does not understand a writer's satirical style of writing, fake news can mislead the opinions of the audience. This issue is a harsh reality for news articles that are posted on social media.

Vosoughi, Roy, and Aral [2] "looked at a highly comprehensive dataset of all the fact-checked rumor cascades that spread on Twitter from its inception in 2006 until 2017. The data include approximately 126,000 rumor cascades spread by about 3 million people over 4.5 million times". Through their research, Vosoughi et al. found that fake news was able to reach 1,500 readers approximately six times faster than the corresponding truths. Additionally, the truth was never spread beyond a depth (i.e. the number of times an original tweet was retweeted by unique users) of ten while rumors were able to reach a depth of 19 in the same amount of time.

According to Hunt and Gentzkow [3], fake news sites receive 41.8% of their referrals or click-throughs via social media platforms. Hunt and Gentzkow believe there are three main reasons as to why social media platforms are ideal for fake news. The first reason is the low and minimum effort needed to distribute their content. With technology continuously growing, it has made the distribution of new articles quite seamless and efficient for content creators to spread their content. The second reason is the presentation of social media. Since mobile usage has increased, it has caused users to view social media on their cell phones. The dimensions of a cell phone screen are smaller than a personal computer, so the viewer's time and focus are often limited. This relationship creates difficulty when reviewing the validity of the news article. The third reason is due a person's personal alignment. Generally, people who have a certain view or opinion will more readily believe other individuals who share similar thoughts, so social media allows individuals to build and collaborate with each other rather than allowing new and progressive thoughts to be created and spoken.

With the significant share of fake news obtained from social media, two problems are created. The first problem is misinforming the public of current events and the second problem is entities or news corporations promoting their own biases. First, misinforming the readers will steer their opinions and ultimately, their decisions towards an undesired path. For example, a student who is applying for universities and colleges could encounter an article regarding a scandal that would dishearten their application for that university. Second, the entities and news corporations who present their content as truthful and objective could promote their views due to hidden bribes or alternative motivations. With all of the creators of fake or biased content, it has caused a shift on how the general public receives news and is informed. Their decision

making is not hindered and misinformed. These forms of fake news have affected critical events, such as the 2016 US presidential election.

Of the 2016 presidential election of the United States, Guess and Nyhan [4] said "Using unique data combining survey responses with individual-level web traffic histories, we estimate that approximately one in four Americans visited a fake news website from October 7-November 14, 2016". One in four - or 25% - of content viewers is a significant portion of the general public. Through their research, Guess and Nyhan found that of these 25%, a larger proportion of those citizens visiting fake news articles were supporters of Donald Trump. However, regardless of a person's political affiliations, the impact of fake news can influence their votes. This concern has motivated us to create a new solution in an attempt to help identify and better inform people.

Fake news can also be a way for a company to unethically bring in viewers or readers for financial or political gain. There are a few reasons why fake news might spread beyond purposeful advantages: the public believes it is real and wants to inform audiences via word of mouth or most likely via social media, reverse someone's opinion on a topic, or utilize the fake news as a form of entertainment - whether they were aware it was valid or falsified.

While all of these reasons can seem harmless, there are negative consequences. One of the main problems of fake news is that most people obtain it over the internet where it can be easily sabotaged. Digital news is growing rapidly and most young people are at risk of ingesting inaccurate information. It is important to discern between real and fake news to better society. News can have a great impact on a population and culture; it can polarize or bring people together. The goal is to classify fake news, examine multiple algorithms for detection, and find the best model to predict with significant accuracy whether a news source is fake.

Fake news is harmful, and misinforms the general public during decision making. Current methods used to detect fake news are inefficient and inaccurate. In this paper, we explore the current and past methodologies used to detect fake news and propose our own method. We describe which features and attributes classify a news source as being "fake" and what is the best method to apply in identifying fake news that is fabricated or biased. Finally, we describe in this paper a proposed detection algorithm that will give a percentage of an article's accuracy and bias compared to articles with a similar topic, a process we are thinking of as reverse plagiarism detection.

2 Detection Algorithms

To better understand the history of how fake news has previously been identified or predicted, several methods are diagnosed. Not only have machine learning algorithms been used to perform this task, but multiple mechanisms of processing, testing, and extracting have been implemented as well. The papers are summarized by their data, method, best approach, and other mechanisms.

There are several different methods of fake news detection, yet none of them are quite what we are looking for. We will propose our idea for reverse plagiarism

detection and describe the algorithm along with the procedure, and then show our results from our testing.

In "Fake News Detection on Social Media: A Data Mining Perspective," a survey was used in order to gather more data on the problem [5]. The summary of these findings is described in Table 1.

Table 1. Summary of key points from Shu, *et al.*

Data	Methods	Best Approach	Other Mechanisms
Survey	Naïve Bayes, Decision Trees, Logistic Regression, k Nearest Neighbor, Support Vector Machines	Unsupervised Machine Learning Models	Knowledge, Style, and Stance-Based, Feature Extraction, True Positive/Negative, and False Positive/Negative

Social media is fairly recent as a news-spreading platform and the authors aimed to display its importance. It introduced both "malicious accounts on social media for propaganda" and "the echo chamber effect". The first type referred to news generated not from humans, but rather bots. The bot was defined by an algorithm that created content that communicated with humans. Examples of bots were most famously acted in the 2016 US presidential election, which utilized Twitter to sway opinions of people who would ultimately vote. Next, rather than a bot that randomly generated content, was the explicit aim of news for selected people. An example of this phenomenon was expressed by certain people adding and growing to become a group of people who all believe the same false news. This effect would spiral into people seeing that news as credible because the others around them shared and propagated it further (i.e. "social credibility"). The other effect lends to "frequency heuristic," which described people believing news content that was false but were more likely to believe it because they heard it so often.

The attempt to describe fake news was either authenticity or intent oriented. Furthermore, methods of detection included knowledge-based, style-based, stance-based, and propagation-based. The data mining framework proposed was feature extraction and model construction. A collection of true positive, true negative, false negative, and false positive results was used in the classification algorithm that produced precision, recall, F1 (precision multiplied by recall then decided by the sum of precision and recall). Similar to what we expound upon is the notion of accuracy as the similarity between predicted fake news and real fake news. Further talk of singular value decomposition and network propagation algorithms were mentioned as well. Some of the general methods examined were semi-supervised, supervised, and unsupervised as well as application-oriented, fake news diffusion, or intervention. While this paper did not aim to seek out a specific algorithm to describe with code, they discuss naïve Bayes, decision tree, logistic regression, k-nearest neighbor, and

support vector machines. Their final conclusion is that unsupervised models should be used because they are more practical as they are more effective in utilizing datasets.

In "Fake news propagate differently from real news even at early stages of spreading," identification of the structure of fake news networks to trace the spread is brought up first [6]. The summary of these findings is described in Table 2. Using Weibo (a Chinese microblogging website) data for collection, a network was created consisting of users and re-postings. Twitter data was gathered specifically about earthquake news in Japan. They associated relevant keywords to the contents of the fake news articles that were already known, such as places and personal names. To determine what may be considered real news data, official accounts with Twitter badges were used to verify the accounts that were most likely to be tweeting real news. From these two facets, a network model was established. The ratio of layer sizes was calculated along with distances of the nodes of the network. Further metrics like probability of fake news and accuracy of predictability were consulted. Using a Welch Two Sample t-test shows whether there was a significant difference in the ratio of layer sizes of fake from real news. All of these metrics were plotted to visually represent the results and to better help the reader understand the complex findings.

Table 2. Summary of key points from Zhao, *et al.*

Data	Methods	Best Approach
Weibo and Twitter Tweets on Japanese Earthquakes	Keywords, Probability, Welch Two Sample t-test	Network Model

In "Fake News Detection with Deep Diffusive Network Model," Zhang, Cui, Fu, and Gouza [7] compared multiple methods and one was developed for production. The summary of these findings is described in Table 3. The paper discussed that there were several issues with the problem of fake news detection. The first concern was the formulation problem. It addressed that there was a lack of formal definition necessary even before the problem was studied. Next, there was the issue of "textual information usage". In order for capture signals to be utilized correctly, there was the need for a feature extraction model (i.e., textual information regarded content and a profile from the social media used). Lastly, there was "heterogeneous information fusion", which brought up the mechanism of correlation. For example, for credibility inference, a relationship between the article-subject and authorship was established. After the matters of the problem were better defined, methods could then be used.

The first method proposed was representation feature learning. More precisely, explicit feature extraction used textual information for signals that helped to determine credibility inference. It was mentioned that not only were certain words shared between fake and true news articles, but also that it was integral to identify the frequency of those words. As a first analysis in labeling, correlations were made. The three facets of the relationships in this method were creators, news articles, and subjects. It was represented that there were explicit features extracted that advanced into a vector

denoting the appearance frequency of the word. This approach was be applied to the creator facet as well.

Table 3. Summary of key points from Zhang, *et al.*

Data	Methods	Their Approach	Other Mechanisms
Creators, News Articles, and Subjects	Propagation, DeepWalk, Line, SVM, and RNN Models	Deep Diffusive Unit Model - "FakeDetector"	Correlation, Explicit Feature Extraction, Latent Feature Extraction, Transformation, Dummy Variables, Hidden Layer, Fusion Layer, Credibility Vectors, Accuracy, P1, Precision, Recall, Network Embedding, Bi-Class Inference, and Multi-Class Inference

The next subcategory was latent feature extraction. Apart from the obvious words that were extracted, there were also somewhat hidden words, which were denoted as hidden signals. The maximum length was critical in calculation of the formula. The feature vector was synced with the words from the article. Because these attributes were words and text, it was best to represent them in a numeric way, which was easier for model performance, efficiency, and simplicity. One of the ways that this transformation could be performed was by one-hot encoding, such as declaring it as a dummy variable, to then utilize it as a binary code. The next model incorporated was the RNN model exploiting basic neuron cells. Therefore, in addition to the hidden layer they also had the fusion layer. Ultimately, this path was fed into a deep diffusive unit model, which is the other method. As described in their earlier analysis, the credibility of these articles was correlated greatly with both the creators and their subjects.

One of the benefits of the deep diffusive unit model in conjunction with the "FakeDetector" was the architecture; multiple inputs could be considered at the same time. To classify the adjustment of the characteristics of the model, additional nodes were created. The last focus on the method was how the model actually learned. The training set of the credibility vectors helped to define the loss function, which ultimately tied into the prediction results. In order to analyze how the model performed, experiments were executed. The accuracy, P1, precision, and recall were assessed. FakeDetector, lp (Propagation), DeepWalk, Line, SVM, and RNN were simultaneously plotted for the aforementioned categories. One of the most common methods in defining fake news in general, not just in the article but in others as well, was the

mention of classification as true, mostly true, half true, mostly false, false, and pants on fire, or some variation of that. The methods, in more depth were compared.

The FakeDetector established that an inference on the credibility labels were used in a GDU (gated diffusive unit) model containing news articles, creators, subjects, in alignment with explicit and latent features. DeepWalk was a model that used network embedding, which embedded the latent feature space from learning. The Line model benefited from being more scalable, using optimized global and local network structures. The Propagation model consisted of labeling as well, but also considered nodes and links; the label-score was used for prediction. Next, was the RNN model used for latent learning into vectors. Last, was the SVM approach, which applied the raw text for explicit feature use.

Evaluation metrics, bi-class inference results, and multi-class inference results were mentioned to conclude, noting that bi-class inference was a better indication of prediction success. To sum, the FakeDetector model created was developed from inspiration of fake news primality work, spam detection research and applications, as well as deep learning research and applications. The deep diffusive network model was proposed, and proliferated into the GDU model resulting in multiple benefits that have proved to demonstrate great performance; more specifically in distinguishing fake news from articles, creators, and subjects.

The two methods chosen to evaluate and tune were logistic regression and multinomial naïve Bayes where we examined a dataset from DataCamp[1]. We chose these two modeling approaches because they work well with binary responses and textual data used in NLP.

3 Reverse Plagiarism Model

Through our research we come to see that current methods of identifying and addressing fake news fall short of the desired outcomes. A new method is needed, and we propose a "reverse plagiarism" detection method to assist in the ongoing battle with false and misleading news items.

In each variation of fake news that we have addressed, the basis of those variations can be boiled down to one thing: real news is based on facts, whereas fake news is not. Due to this, we hypothesize that in a selection of articles on the same topic, all real news articles on the topic will resemble one another to some degree while the fake news article will stand out. It will stand out because of the "alternative" facts it describes that are not present in any other articles on the same topic.

It is these facts that led us to the consideration of what may be considered "reverse" plagiarism used to detect fake news. Our thought is this: if real news articles resemble one another due to use of the same facts, there should be some degree of resemblance to classic plagiarism. As such, we believe that using plagiarism detection methods to find articles that, to a plagiarism detector, seem to be original will lead us to discover news items that are *too* original—or fake.

[1] https://s3.amazonaws.com/assets.datacamp.com/blog_assets/fake_or_real_news.csv

Using Python in Jupyter Notebook with the Pandas library, we read in files that are imported from the webhose.io API[2]. A query is established that works to filter certain words from articles. This API reads in data from web content with four different formats of JSON, XML, RSS, and Excel and transform the content into machine-readable data. The time range can be adjusted so that we can increase our sample size, and ensure that we are obtaining historical data from archives and not simply the most current information. The data is then imported using the Python code.

The resulting output prints the raw text of the news articles. Using the difflib function from the SequenceMatcher library[3], a similarity ratio was generated between any articles that are opened in order to detect plagiarism. The class is used to compare type sequences for hashable elements. This algorithm is based off of the Ratcliff and Obershelp approach. The specific class of difflib is a general method of differing. It works by comparing the line sequences with similar characters. The ratio function will return the relationship between a range of zero and one; one meaning that the texts compared are completely the same, and zero meaning they are completely different.

We work to refine this algorithm and instead of using it to detect plagiarism, we can define the relationship of various text files from news articles that will be classified as fake. In our case, we want the articles to appear to the algorithm as plagiarism. This is because our hypothesis is that the more the articles appear to be plagiarized to some degree, the more likely they are to be based upon the same true facts and not contain false or misleading information. When outliers are found, such as those that have a low ratio, they would be identified as an article that has different text but with the same topic.

In our solution, we dissect what is occurring by these functions on a coding level and compare our ratio. We already knew which news articles are fake or not, therefore we could distinguish whether ours is the former or the latter. Additionally, since this function that already exists is not explicitly for testing fake news, but for plagiarism, ours was applied more directly and more efficiently.

4 Webhose.io

The webhose.io website utilizes data feeds from various websites or domains. Its main process is to select the website, publisher, date, and number of articles from the web. The grander use of this site is to ultimately give access to large amounts of data at scale, for products like machine learning algorithms on either the academic or professional realm. Several tabs include ‘Dashboard’ and ‘API Playground’. A query is performed consisting of multiple filters, such as: ‘Format’, ‘Crawled Since’, and ‘Sort By’. This method helps to run a more specialized query for a given problem statement or need for data.

In order to use the data within Jupyter Notebook, integration of a coding language is needed. Python is the language of choice due to its popular and useful libraries. Two of the most import libraries imported are Pandas and NumPy, usually portrayed in the code as ‘pd’ or ‘np’ for shorthand notation. These libraries allow the text files to be

[2] https://webhose.io/web-content-api

[3] https://docs.python.org/2/library/difflib.html

converted into dataframes. Once they are in that format, they can be worked to create plots and descriptive statistics for exploratory data analysis.

The many ways to filter the query are shown in Figure 1. It also prints out an 'Endpoint' URL for the Python code to reference when selecting the article.

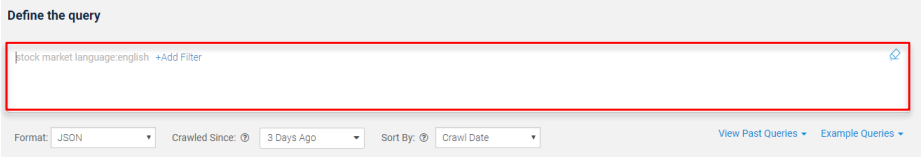

Figure 1. The webhose.io main query page

Once the process is understood, the next step is to acknowledge the facet to use for the project. Our dataset includes articles from weather events. Therefore, in our filter we use 'Hurricane Florence' as our common denominator between articles so that comparing the similarity ratio of the texts makes sense. Other types of articles to filter down to could be along the line of politics, sports, education, health, energy, international events, etc.

The importance of natural language processing (NLP) within these libraries and methods is substantial. The reason this tool is so widely utilized is due to its benefits of large-scale metrics and analysis on worded and categorical data. Being that the articles are mainly words and/or strings, the process of tokenizing them is one of the main uses of NLP. It allows the algorithm from SequenceMatcher to breakdown the mean, median, and modes of word occurrences. It also allows for redundant words to be deleted in the function so that there is not misleading data and results, or false positives and negatives.

In order to dissect the specific article, it is important to label the 'post' number, 'publication date', 'thread', and 'site'. These unique identifiers pull the article with those associated attributes. The vocabulary size of the articles may range so it is critical to normalize the amount when comparing ratios. The 'lexical diversity' is another important tool when analyzing text files to compare the length of the text to another. To use the lexical diversity function, the 'len' function is needed to create a number of the amount of words contained in the article, whether that be from the title/header, body, or author.

The NLTK library is further tool that tokenizes the text files. It also has the ability to create a lexical dispersion plot where a list of chosen words can be plotted on the y-axis and the word offset is plotted on the x-axis. To further diagnose where these words are in the text files that are being compared, the 'concordance' feature is used to print out a list of text where the location of that word is with a contextual example.

All of these tools help to explore the data or text files from the articles before using them in the SequenceMatcher algorithm. The next step is therefore to assign those text files a unique identifier so then the similarity ratio of them can be compared.

5 Logistic Regression and Multinomial Naïve Bayes: Results

Between the logistic regression and multinomial naïve Bayes, the logistic regression model performed better than the multinomial naïve Bayes in terms of the accuracy and area under the curve (AUC) score. The accuracy score of the logistic regression model was 90.9% with an AUC score of 0.963 and the multinomial naïve Bayes model had an accuracy score of 89.3% and AUC score of 0.927.

In terms of both accuracy and AUC, the logistic regression model was able to predict more correctly over the total predictions compared to the multinomial naïve Bayes. These results were compared to our plot in Figure 2 below. The graphical plot below is a receiver operating characteristic (ROC) curve. The ROC curve is a visual representation of the range of probabilities.

On the y-axis on Figure 2, it represents the true positive rate of identify a news article correctly. On the x-axis, it represents the false positive rate or news articles that the model classifies are being true or authenticate but the article was actually fake. For this ROC curve, the AUC score was 0.87 based on the range of probabilities. Comparing the ROC curve's AUC score to the logistic regression and multinomial naïve Bayes' AUC score, the two models both performed better than the ROC curve.

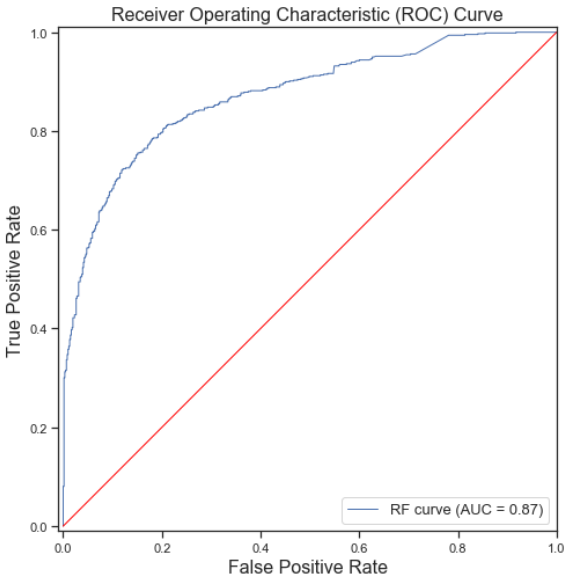

Figure 2. Receiver Operating Characteristic (ROC) Curve

The visualization of the ROC curve and true positive rate, as well as false positive rate can be easily visualized in Figure 2. Both models were able to predict more accurately versus the ROC curve's AUC score of 0.87. After closer analysis, we chose the logistic regression values to be plotted in our confusion matrix in Figures 3 and 4.

The confusion plot below shows the performance of the logistic regression model based on its predictions of true positives, false positives, true negatives, and false negatives. The idea of these matrixes were to better understand the performance with values that were overlaid on a heat map true and predicted labels as a visualization.

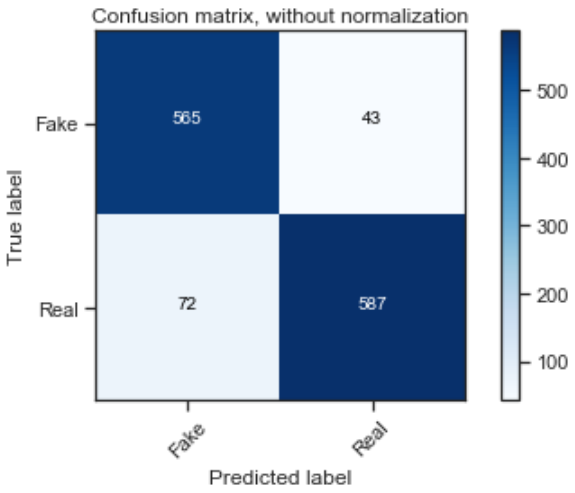

Figure 3. Confusion Matrix (un-normalized)

In Figure 3, the values will represent the logistic regression model's predictions. It is important to keep in mind that these values were utilized without normalization. The total number of the predictions is 20% of total data points. The majority of the model's predictions were in the top left and bottom right quadrant. The top left quadrant represents what the model predicted to be fake and the article being labeled as fake. On the other hand, the bottom right quadrant represents what the model predicted to be real and the article being labeled as real. These values are better illustrated in Figure 4.

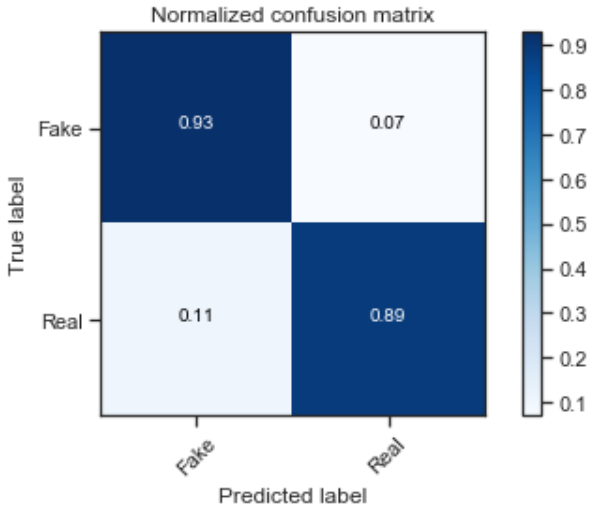

Figure 4. Confusion Matrix (normalized)

Figure 4 is a confusion matrix that describes the values after normalization. Notice how the scale on the right side of both figures show a different range before and after the normalization of values. After normalizing the values, it brings the range of values to be narrower to fit between 0 and 1. As mentioned above, the logistic regression model was able to identify 93% of fake news and 89% of real news. The model did not classify 7% of the fake news and 11% of the real news correctly.

6 Reverse Plagiarism Model: Results

For our initial testing, our group analyzed two files that related to Hurricane Florence. The files come from the webhose.io API which uses a filter based on the date, site type, and subject matter. Using the integrated code generator, we utilized the code, as shown in Figure 5, to create a text output based on our parameters. After running the Python code, we then converted the article bodies into plain text files.

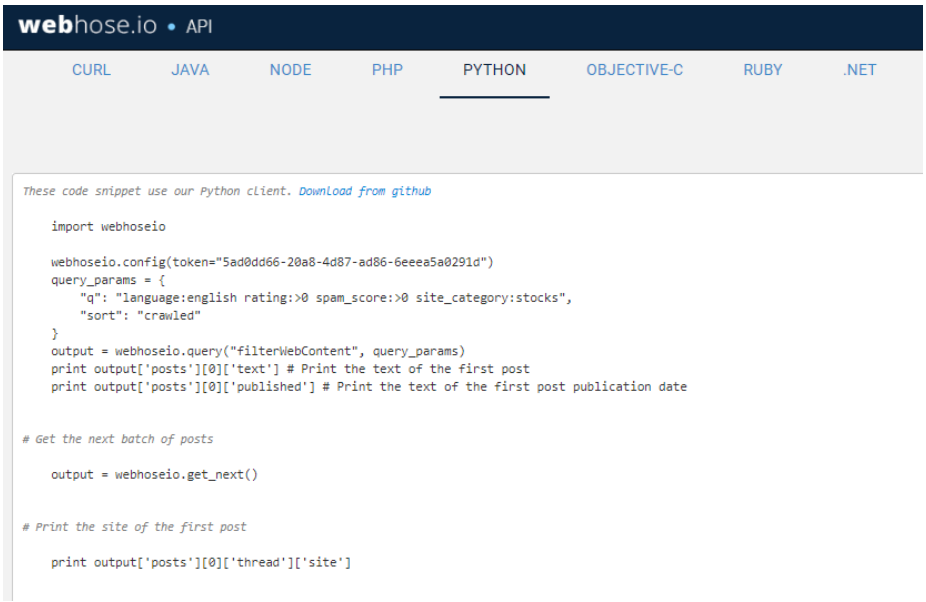

Figure 5. The integration code of Python in webhose.io

The next step, after importing the text file, is displaying that text in our Jupyter Notebook using Python. It prints out the full data from that article. As shown in Figure 6, the Python output gives the body of the article, date of the publication, and publisher.

The files were compared using Python libraries and functions. For our analysis, we will use the difflib library and SequenceMatcher module within the Python environment.

```
Hurricane Florence put a corridor of more than 10 million people in the crosshairs Wednesday as the monster storm closed in on
the Carolinas, uncertainty over its projected path spreading worry across a widening swath of the Southeast.
Faced with new forecasts that showed a more southerly threat, Georgia's governor joined his counterparts in Virginia and North
and South Carolina in declaring a state of emergency, and some residents who had thought they were safely out of range boarded
up their homes.
The National Hurricane Center's best guess was that Florence would blow ashore as early as Friday afternoon around the North Ca
rolina-South Carolina line, then push its rainy way westward with a potential for catastrophic inland flooding.
Florence's nighttime winds were down to 110 mph from a high of 140 mph, and the Category 3 storm fell to a Category 2, with a f
urther slow weakening expected as the storm nears the coast. But authorities warned it will still be an extremely dangerous hur
ricane.
2018-09-13T07:33:00.000+03:00
breitbart.com
```

Figure 6. An example print out of the text file displaying the title, body, date, and domain

In the SequenceMatcher module, we used a function called SequenceMatcher. The SequenceMatcher function compares the similarity of the text between the two files and gives a similarity ratio. A low value represents a large difference between the two articles or lack of plagiarism. On the other hand, a high value represents a small difference or significant chance of plagiarism. Our initial results show a similarity ratio of 1.234%, which represents that the two files are very original and lack evidence for plagiarism as shown in Figure 7.

```
# Analyzes the similarity of text between two files
from difflib import SequenceMatcher
with open('file1.txt') as file_1,open('file2.txt') as file_2:
    file1_data = file_1.read()
    file2_data = file_2.read()
    similarity_ratio = SequenceMatcher(None,file1_data,file2_data).ratio()
    print(similarity_ratio)  #plagiarism detected
```

```
0.012342748635176834
```

Figure 7. First run of different text files using SequenceMatcher

After reviewing the text of the two files, we noticed that they were related in the same subject matter but speaking on two different concerns. The first article talked about the severity of the hurricane while the twentieth article spoke about the effects it will have on the price of pigs. Both articles could be valid and accurate but for our testing, it proves to be a false positive in terms of a "fake article".

When using the SequenceMatcher function to analyze text files of similar themes in weather like "Hurricane Florence", we found that it is only useful for that one facet. We hope to expound upon this similarity ratio for other types of articles so that our classifier of fake or real labeling can be utilized for a greater set of data.

In the future, we will use this new finding to refine our testing and model. We plan to increase our testing to include more articles and compare their similarity with one another. After analyzing the similarity of the articles, the approach can be applied to new and current articles. Perhaps other similar methods could be combined with our reverse plagiarism model to compare articles at scale with a ranking system. We hope that this methodology will yield more accurate results and findings in the future.

7 Ethics

In order to obtain models that could be robust and accurate, we needed to accumulate a large volume of data. We partnered with the webhose API in order to make sure the data we were ingesting was not taken without permission. Furthermore, we utilized a public dataset in our logistic regression and multinomial naïve Bayes classification models from the website DataCamp which allows downloads of thousands of files of data. We wanted to ensure that our data was obtained correctly and ethically.

As data scientists, we understand the importance of ethics on personal data. If storing data is important to a data science project, it is expected to refresh and anonymize the unique identifiers or to simply use public data that is allowed for science and exploration. Our results and findings are also public and can be accessed by anyone with a computer. The goal as a scientist is to always learn, thrive, and be a part of a community that feeds off one another. Not only is it respectable to follow ethical practices in obtaining and using other data, but also in collaborating and presenting data and findings to wider audiences.

In a different sense, our project itself sheds light onto the repercussions of knowingly posting fake news on a website. In the future, our model could be used to score, rank,

and help identify which articles are fake or not and can help to bring ethical soundness to thousands of websites carrying false claims. One of the main problems with fake news is that it is harmful; it is a main part of this paper to follow ethical guidelines in our own modeling. Furthermore, it is also essential to help reveal fake articles with the benefit from data science.

Regarding the harmfulness of fake news itself, it can be considered detrimental to a reader. On several social media sites, blogs, and news sites, articles can circulate quickly and can contain propaganda, racism, sexism, and many more unfortunate implications. Propagating fake news brings up a complex question: is it the person posting, the writer, or the site where the article is posted at fault? The main players in this media frenzy are Google, Twitter, and Facebook. Will these large companies face lawsuits? With technology expanding and social media becoming the forefront of our thinking and everyday life, it is up to everyone to ensure that they are reading the source material cautiously and to report any suspicious or hateful wording that could suggest fake news.

Data science can be used to back up or promote these fake news articles in circulation. There is a good and bad side to the algorithms that monitor and edit materials before being published to the whole internet. Someone should be responsible, but it is in everyone's hands to act on how they can deter fake news from spreading. If you are a website, it would be intelligent, safe, and thoughtful to evaluate all of your articles – this is where responsibility comes in. Although the site has thousands of articles, it is not an excuse to keep circulating them. Machine learning should be used to identify fake news and perhaps create a survey for every user to identify if they think it is fake news. Perhaps logistic regression, multinomial naïve Bayes, and plagiarism detection algorithms could be utilized to stop the spread of fake news.

8 Conclusions

Classifying fake news was proven to be easier than originally thought. The main steps of identifying fake news were: collecting the data, cleaning or vectorizing the text data, analyzing the cleanse data, and producing an outcome from the analysis. Gathering news articles to a usable format took a substantial amount of time. Webhose.io's query search engine allowed the data extraction to be much more consist and efficient based on the analysis. Analyzing the key features, such as the body of the article and title, proved to be the most useful part for the analysis. Those text features need to be properly separated by vectoring each word in order for the model to compare each instance of the word against the other text documents. Once the dataset was collected and cleaned, the analysis was the next step.

The proposed reverse plagiarism method proved to be quite time consuming and difficult to scale once the size of the data increased. As a result, the remaining amount of time was dedicated to building and creating the traditional classification for identifying fake news. By analyzing the two other models, logistic regression and multinomial naïve Bayes, the accuracies were both higher than 90%. Once again, those results prove that fake news can be easily identified, but more importantly that the analysis of text documents can be scaled and evaluated much faster than a human could and can eventually be used to rank or score news articles on websites.

In terms of the next steps, other forms of processing the text data could be utilized. Performing a document similarity analysis would greatly help with the time restriction that hindered the performance of the reverse plagiarism method. A more advanced form of document similarity analysis, cosine similarity, would ideally be incorporated as a suggested area of improvement.

There has been some progress made with the approach and analysis but there is room for more growth. With extra time, more data should be collected with the goals to create a more diverse data population of content creators and increased the number of articles used for the analysis. The increase of data size would help represent scope of fake news in the current status and help future analyses. Currently, the logistic regression model outperformed the other traditional classification method and the proposed method, reverse plagiarism. For future analysis and recommendations, the audience should focus on titles and sources of news articles to make sure they are not reading content that is inaccurate or false.

References

1. University, Benedictine. “What Kinds of Fake News Exist?” Fake News: Develop Your Fact-Checking Skills, 2017, researchguides.ben.edu/c.php?g=608230&p=4219633.
2. Vosoughi, Soroush, et al. The Spread of True and False News Online. Science, 2018, pp. 1146–1146, The Spread of True and False News Online.
3. Allcott, Hunt, and Matthew Gentzkow. “Social Media and Fake News in the 2016 Election.” Https://Web.stanford.edu/~Gentzkow/Research/Fakenews.pdf, 2017.
4. Guess, Andrew, and Brendan Nyhan. “Selective Exposure to Misinformation: Evidence from the Consumption of Fake News during the 2016 U.S. Presidential Campaign.” Https://Www.dartmouth.edu, 8 Jan. 2018, www.dartmouth.edu/~nyhan/fake-news-2016.pdf.
5. Shu, K., Sliva, A., Wang, S., Tang, J., and Huan, L. (2017). “Fake News Detection on Social Media: A Data Mining Perspective”. arXiv:1708.01967v3.6.
6. Zhao, Zilong, et al. (2018). “Fake news propagate differently from real news even at early stages of spreading”. arXiv:1803.03443v2.
7. Zhang, J., Cui, L., Fu, Y., Gouza B., F. (2018). “Fake News Detection with Deep Diffusive Network Model”. arXiv:1805.08751v1
8. FÂRTE, Gheorge, & Daniel Rareș OBADĂ. "Reactive Public Relations Strategies for Managing Fake News in the Online Environment." Postmodern Openings [Online], 9.2 (2018): 26-44. Web. 12 Jun. 2018
9. Berinsky, Adam J. "Rumors and Health Care Reform: Experiments in Political Misinformation." British Journal of Political Science 47.2 (2017): 241-62. Print.
10. Zubiaga A, Liakata M, Procter R, Wong Sak Hoi G, Tolmie P (2016) Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. PLoS ONE 11(3): e0150989. doi:10.1371/journal.pone.0150989
11. Jarmul, Katharine. “Detecting Fake News with Scikit-Learn.” Detecting Fake News with Scikit-Learn, DataCamp Community, 24 Aug. 2017. www.datacamp.com/community/tutorials/scikit-learn-fake-news.
12. “Ethics in the News - Fake News and Facts in the Post-Truth Era.” Ethical Journalism Network, Ethical Journalism Network, ethicaljournalismnetwork.org/resources/publications/ethics-in-the-news/fake-news.
13. Docketrun. “Docketrun/Detecting-Fake-News-with-Scikit-Learn.” *GitHub*, 1 Jan. 2018, github.com/docketrun/Detecting-Fake-News-with-Scikit-Learn/tree/master/data.
14. Huang, Anna Y.-Q. “Similarity Measures for Text Document Clustering.” (2008).
15. Ezeiza Alvarez, Jon. “A review of word embedding and document similarity algorithms applied to academic text.” (2017).
16. Elsafty, Ahmed. “Document Similarity Using Dense Vector Representation. (2017).
17. Tom Kenter and Maarten de Rijke. 2015. Short Text Similarity with Word Embeddings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15). ACM, New York, NY, USA, 1411-1420. DOI: https://doi.org/10.1145/2806416.2806475