Fake News: A Study of the Differences between Real and Fake News

Final Report

Data Mining – CSPB4502

Group 3: Claudia Hidrogo, Madeline Odom, Jordan Sims, and Xiomara Winkler

1. Abstract

According to the New York Times, “We are in an era of endemic misinformation -- and outright disinformation”, (Fisher, 2021). One very important facet in the battle against misinformation is the vital step of identification. As time goes on, it appears that misinformation is becoming harder and harder to spot to the naked eye. The goal of this project is to study AI-identified fake news, as well as real news, to determine key differences that could help the average newsreader be able to identify the differences between fake and real news.

In our research, we sought to answer the following questions: Is there an association between length of article and the realness of news? What words are most commonly used in real vs fake news? Is there a time when more fake news started showing up (in the Harvard set)? What categories (sports, politics, entertainment) are more prone to fake news? What publications are more prone to publish fake news?

We found that the average reader will not be able to deduce if an article is fake based on article length or word choice. Additionally, we found that political news tends to be most prone to producing fake news and that fake news tends to be posted more frequently during election cycles in the United States. In our research, we also discovered that certain publishers of news generate more fake news than others.

2. Introduction

In our efforts to process and classify real and fake data, we found these questions important in helping us determine if there were any key identifiers of fake news. For example, fake news is commonly associated with political news; however, we wanted to account for various news sources in our research to determine which categories are most prone to fake news. Similarly, we wanted to identify if certain publications were responsible for publishing fake news, which could potentially help us identify biases in the media. Another question we wanted to explore involves the timing of fake news which is important because it can help us identify if there are any associations with the frequency of fake news during key historical events. Finally, we wanted to explore the construction of fake news media by determining if there were key words that were commonly used in fake news and if length of the article plays a factor as well. These questions shaped our research and revealed valuable insights that are discussed further in this paper.

3. Related Work

Fake news and misinformation have been topics of growing conversation over the past few years, and the field of study on this topic has been growing as well. This interest increase has led to multiple research papers and machine learning models like “Fake News Detection Using Machine Learning

Approaches", authors Khanam, Alwasel, Safari, and Rashid, which discusses advanced methods of building AI machine learning models that can effectively differentiate fake news from real news. Another example is "Fake News Detection On Social Media: A Data Mining Perspective", authors Shu, Silva, Wang, Tang, and Liu's, where they take a similar view to the fake news detection. Additionally, Kajal Kumari utilizes a machine learning model to accurately classify news data as real or fake in their paper, "Detecting Fake News with Natural Language Processing" as does Ben Roshan in his piece "Fake news classifier LSTM." These works focus on the problem from the perspective of computer science, which is decidedly different from the objective of this project.

4. Data Set

Title	**Source**
A Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles	Harvard Dataverse
Dataset consisting of fake and real news articles.	Kaggle: Clément Bisaillon
Fake News Detection - Train dataset	Kaggle: Samrat Sinha
Classifying the Fake News – Training dataset	Kaggle: Paul Larmuseau and Deeachain
Detecting Fake News with Python and Machine Learning	The Fake News Dataset
Generic word embedding dataset for fake news detection consisting of real and fake news from popular news sources.	Zenodo: Radu Prodan, Prateek Agrawal, Pawan Verma

5. Main Techniques Applied

5.1 Data Pre-processing

To start our project, our group gathered multiple data sets from across a variety of sources (See Section 2. Datasets). Each of these datasets were downloaded as its own separate CSV file. These files were manually verified in Excel to make sure they all contained the attributes: title, text, and label (specifying whether the article is fake or not). Afterwards, a Python module was created to start organizing the data more neatly, and the Pandas library was used to help in the process of extracting, transforming, and loading the data integration of the files.

5.2 Data Cleaning

The main phase was to prepare the individual data sets and clean them individually before attempting to do the data integration, this included multiple steps.

In the categorical fix step, we made sure that all the datasets only displayed three columns as several of the other empty columns were initially included in the extraction. After fixing the dataframe layout, we proceeded to work on the label attribute, as many of the articles were pre-classified in the source as fake news or true news but did not have a column containing this information, or others were labeled with wrong string type like 'FAKE' or 'REAL'. This column intends to answer the question "is it fake?", so it was necessary to convert or add to all the rows the binary representation of each news article with 0 meaning the article is true and 1 that is fake. Lastly, some of the rows were missing data, so we deleted these rows.

5.2 Data Integration

After performing the data cleaning of each dataset, we preceded to integrate the following files to obtain the master dataset:

Kaggle: Clément Bisaillon as True and Fake

Kaggle: Samrat Sinha as File1

The Fake News Dataset as File2

Kaggle: Paul Larmuseau and Deeachain as File3

Zenodo: Radu Prodan, Prateek Agrawal, Pawan Verma as File4

This master dataset was furthermore cleaned by dropping any NaN values that were still present, and by adding a new column labeled 'contents'. This new attribute contains the title and text information combined to facilitate the data transformation and reduction process.

5.3 Data Transformation

In this step, we will be manipulating the contents attribute of our master dataset by identifying and removing the words from every news article that do not have a significant meaning. The list of words were found by using "stop words" from the Natural Language Tool Kit (nltk) library (See Section 6.6 : Natural Language Tool Kit). Stop words are words commonly used that many programs and projects (such as ours) are programmed to ignore as they do not add any value or context to the text (*Removing stop words with NLTK in python* 2021). Some examples of these stop words are "the", "an", "because", and "of". As result, we obtained the 'clean contents' attribute, this column of string was converted to tokens (numbers) for easy data manipulation. This is known as tokenization.

5.4 Data Reduction

The purpose of this step is to obtain a smaller volume of the dataset, while keeping the veracity of the contents. The padding was performed by using the Tensorflow and Keras (Section 6.5 and 6.7) libraries and its fixed length option.

5.5 Fake News Model

The fake news detection model is based on the project "Fake News Detection with Machine Learning". This project applies deep learning by using tools like Bidirectional Long-Short Term Memory (LSTM) with Tensorflow; it is implemented to be able to establish long term dependencies, these modified Neural Networks allows to better process and correlate related semantic and syntactic features.

The model was applied to a master dataset with a total of 164,768 news articles. It was

run with an epoch of 1 due to its large size (this refers to the number of times an algorithm is going to run), with this data, we obtained an accuracy of 0.85 and a total of 114,538 unique words. These results have been associated with a confusion matrix (See Section 8) to allow one to quantifiably evaluate the results.

6. List of Tools

The programming language Python will be the primary tool used to perform the data preprocessing of the news articles. The following Python libraries will be imported to create, train, and analyze the model.

6.1 Pandas

“Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.” (*Pandas*) We will be using it for its data frame manipulation and analysis tools.

6.2 Numpy

A power tool built for Python with a wide variety of mathematical applications. (*NumPy*) We will be using it for numeric analysis.

6.3 Matplotlib

“Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.” (*Visualization with python*). We will be using it for data visualization in our project.

6.4 Plotly

“Plotly's Python graphing library makes interactive, publication-quality graphs.”(*Plotly python graphing library*) We will be using it in our project to create interactive plots.

6.5 Seaborn

It is a data visualization tool used for statistical graphs. Seaborn will be used for the confusion matrix representation.

6.6 Keras

Keras “offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.“ (*Keras*) is used for deep learning in AI models.

6.7 Natural Language Toolkit for Python

This is a library made for Python that is intended “to work with human language data” (*NLTK*).

6.8 Tensorflow

Tensorflow is an open-source platform for machine learning that enables users to build and train ML models using APIs like Keras (*Tensorflow*).

6.9 Other Tools

PostgreSQL and Heroku tools were used to host primarily the Harvard dataset. This cloud option enabled us to make any changes or updates to the data once without the need to individually update all members’ systems, while also allowing us to back up our database as well.

We created a PostgreSQL database instance on Heroku’s platform with rotating credentials. We used the console to interact directly with the database and set up

administrative permissions, as well as create user accounts.

After testing each member's accessibility to the database, we had to decide how we want to organize the data sources in a relational schema. This database was created to act as a repository for our data sources, therefore it was resolved that we should strive to keep the sources as untouched as possible within the schema, as most of the cleaning was performed prior to the data integration step.

Each table in our database is a member's contribution to the data repository. That is to say, each table is a cleaned and preprocessed data set conducted by a member of the group. To facilitate migration of each member's local data set to the repository a python script was created that would first check to see if the data source already exists, and if not would create a table to store the data source. The data sources were in the forms of CSV files, Excel files, and SQLite databases.

The data type of each column would be based on the python data type that most accurately reflects the data attribute, e.g. a number with a decimal point that would be a float in python would be set as a decimal type in PostgreSQL.

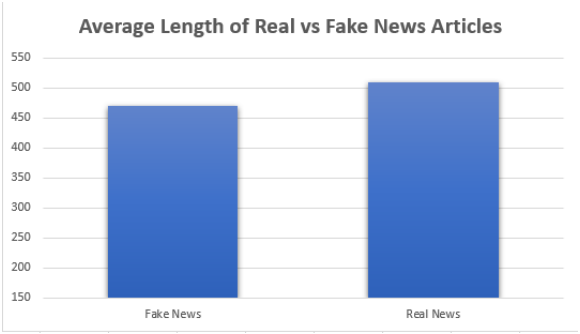
We used Psycopg2 as our ORM to adapt python into SQL commands to interact with our database. First, we used a python script to insert all the data into tables, and then afterward Psycopg2 was used to pull data in our python programs. From there we could easily manipulate the data as it allowed us to integrate it with other data tools, such as Pandas.

7. Key Results

Throughout the project we used similar methodologies from the related work sources (See Section 2); we analyzed the real and fake articles from the datasets to answer a few questions.

7.1 Question 1

Our first question aimed to tackle the length of fake new articles vs real news articles. To accomplish this, we wrote a python script that took the length of all of the articles of both types of articles, and then calculated the average. The goal of this was to determine if fake news articles drastically differ in length from real news articles. The results were that real news articles average around 510 words per article, whereas fake news articles average around 470 words per article.



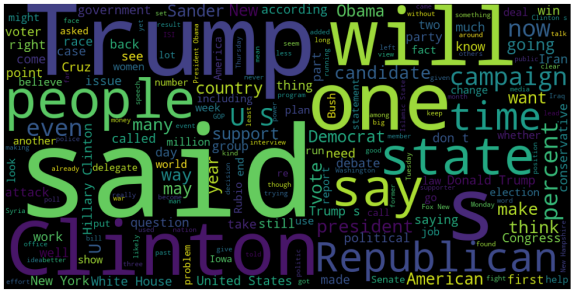
This difference in sizes only accounts for about an 8.5% difference, which, while not being insignificant, is not a wide enough difference for the average news reader to be able to determine just by looking at an article. We have determined that article length, at a glance for the average reader, is not a suitable way to determine if an article is fake or real.

7.2 Question 2

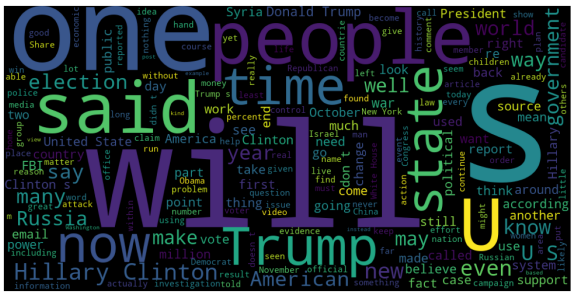
Our second question was to determine if there was a large variation of the words

used in real news articles vs fake news articles. We did this by taking a subset of our data and creating a word cloud from that subset of articles. Below are the results:

Real News Wordcloud



Fake News Wordcloud



From this, we can see that there is little difference between the words used in a fake news article and a real news article. This also will not be a good way to tell between the two for the average reader.

7.3 Question 3

Our third question sought to address external factors to the content and titles of fake news articles. We took a look at our data and conducted some exploratory analysis with visualizations.

We queried our database and created new features based on a current column value, and we also extrapolated subset values.

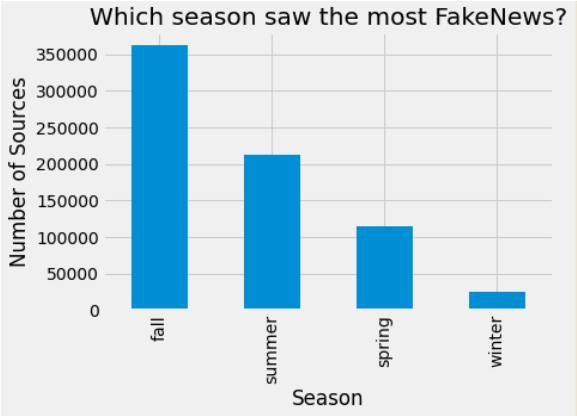
The date column was split into year, month, and day columns so that we could drill down into the data to detect any distinct patterns.

We visualized the counts of year, month, and day respectively, and were able to identify a striking trend. As the month feature advanced towards the end of the year, the amount of fake news articles increased.

In order to verify the trend a new column attribute was created, the ‘season’ column. We created this column by mapping Months to their numerical representation; each season is defined by the months that generally compose them.

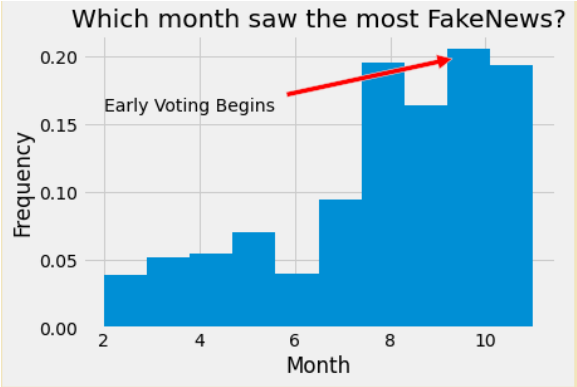
Season	Months	Numerical Representation
Winter	Dec, Jan, Feb	12, 1, 2
Spring	Mar, Apr, May	3, 4, 5
Summer	Jun, Jul, Aug	6, 7, 8
Fall	Sept, Oct, Nov	9, 10, 11

Once the counts were plotted by season we could see that there was a dramatic increase in the number of fake news articles published as we approached the fall season.



Now that the overall trend has revealed a definite effect on the number of fake news articles, we could drill deeper to see if we could make further discoveries.

Our monthly analysis seemed to support the idea that fake news articles tend to be published in the fall. However, not only did this analysis support this claim, but it further indicated that there was indeed a correlation between the publishing of fake news articles and the proximity of the November election.

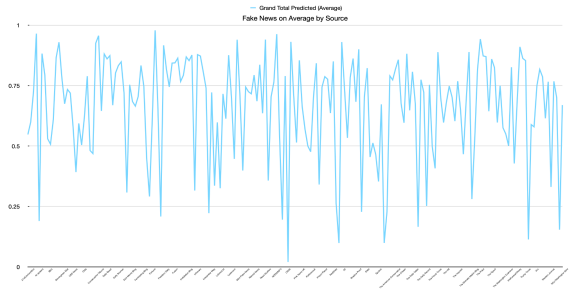


7.4 Question 4

The fourth question we sought to answer asked if there is a particular category, e.g., sports, politics, entertainment, that are more prone to fake news. To answer this question, we intended to use the datasets found; however, many of the sources did not directly label each news article, but instead, labelled the news source as reliable or essentially not reliable. Furthermore, our other datasets did not label the category of news, but they did label whether it was true or false. Thus, it was difficult to find an option that would be able to determine whether fake news was more likely to be of a certain category. On the other hand, from a subjective determination, it appears that the political news was usually the news that was falsified. This result can't be directly linked with any data because of the missing labels, but the observations can be supported by the idea that political articles tend to be the news that has the most reward for false labelling; thus, through both observation and reasoning, we can deduce that political news has the greatest chance of being falsified.

7.5 Question 5

The fifth question we sought to answer asked if certain publications produced more fake news than others and we discovered they do. To investigate this question, we utilized "A Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles" from Harvard which includes over 500,000 articles across approximately 192 publications. This robust dataset includes well known media sources such as CNN and BCC as well as smaller outlets like Addicting Info and Freedom Outpost.



We used a Naive Bayes Classifier to process this data and determine real and fake news predictions. Our predictions marked fake news as ‘1’ and real news as ‘0’. We calculated the average of the Real and Fake predictions per publication to measure the frequency of the real and fake predictions as the publications representation varied in the dataset. The publication, “Forward Progress” produced the maximum average of fake news at 0.9788 whereas “OSCE” produced the minimum average at 0.0201. More well-known publications such as BCC averaged 0.507, CNN averaged 0.621, Buzzfeed averaged 0.719 and Infowars averaged 0.878.

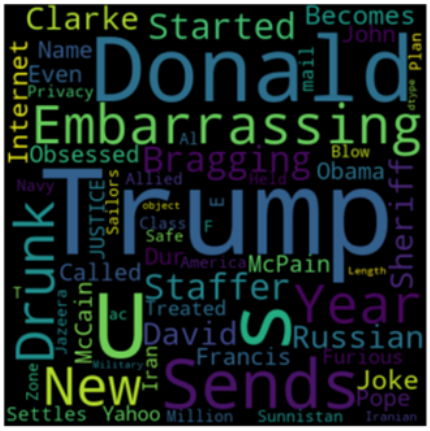
8. Data Visualizations

In the initial steps of the data processing, we can use the contents of a fake news article in a word cloud visualization to help pinpoint the most common words found. An example of this is the following word clouds based on the fake dataset from [Kaggle: Clément Bisaillon.]

The images are organized in three groups: 1) text alone, 2) titles alone, and 3) text and titles. This was created with the idea of investigating the importance of including both the titles and the text versus just one alone. This allows us to better understand the nuisances involved in the machine learning algorithm to detect fake news.

Article Titles

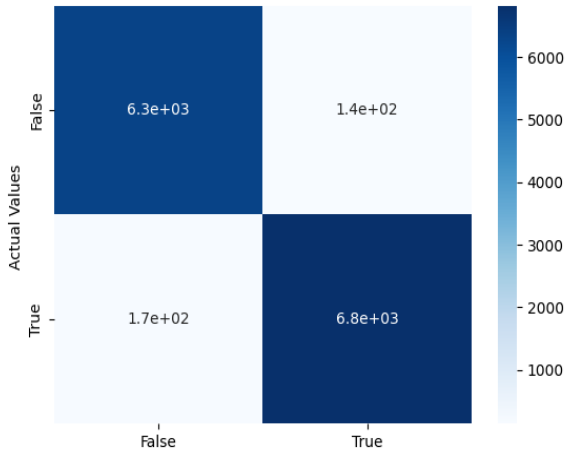
Text Content



Article Titles and Text Content

Another data visualization example is the confusion matrix, it is used to view the performance of a classification model. In

this project, we used the Fake and True dataset (See Section 5.2) to determine the true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). As result, we obtained 13.476 of actual and predicted values.



9. Applications

In conclusion, we discovered some key insights about fake news that may benefit a news consumer. First, the average reader will not be able to deduce if an article is fake based on article length alone. Second, the average reader will not be able to deduce if an article is fake based on word choice alone. Additionally, fake news articles tend to be posted more frequently during election cycles in the United States. Political news is by far the news category most prone to fake news. Last, it is clear that certain publishers of news generate much more fake news than others, so it is important when getting news to verify the integrity of the publisher before trusting what you read.

10. Sources

Aruchamy, Vikram (2021, September 19). *'Confusion Matrix in Python'*. Retrieved December 6, 2021, from https://www.stackvidhya.com/plot-confusion-matrix-in-python-and-why/.

Christian Versloot (2021, January 11). *'Bidirectional LSTMs with TensorFlow 2.0 and Keras'.* Retrieved November 10, 2021, from https://www.machinecurve.com/index.php/2021/01/11/bidirectional-lstms-with-tensorflow-and-keras/.

Coursera. Ahmed, Ryan *'Fake News Detection with Machine Learning'*. Retrieved October 5, 2021, from https://www.coursera.org/projects/nlp-fake-news-detector.

E. Bender and A. Lascarides. *Linguistic Fundamentals for Natural Language Processing II: 100 Essentials from Semantics and Pragmatics*. Morgan & Claypool, Toronto, 2019.

Fisher, M. (2021, May 7). *'belonging is stronger than facts': The age of misinformation*. The New York Times. Retrieved October 24, 2021, from https://www.nytimes.com/2021/05/07/world/asia/misinformation-disinformation-fake-news.html.

Keras Team. (n.d.). Keras. Retrieved October 24, 2021, from https://keras.io/.

Khanam, Z., Alwasel, B. N., Sirafi, H., & Rashid, M. (2021). Fake news detection using machine learning approaches. *IOP Conference Series:*

Materials Science and Engineering, *1099*(1), 012040. https://doi.org/10.1088/1757-899x/1099/1/012040

Kumari, Kajal. (July 19, 2021). *Detecting Fake News with Natural Language Processing.* Analytics Vidhya. Retrieved December 4, 2021. https://www.analyticsvidhya.com/blog/2021/07/detecting-fake-news-with-natural-language-processing/

Mushtaq, Sana (2019, June 14). *'Data preprocessing in detail'.* Retrieved November 10, 2021 from https://developer.ibm.com/articles/data-preprocessing-in-detail/.

NLTK. (n.d.). Retrieved October 24, 2021, from https://www.nltk.org/.

NumPy. (n.d.). Retrieved October 24, 2021, from https://numpy.org/.

Pandas. pandas. (n.d.). Retrieved October 24, 2021, from https://pandas.pydata.org/.

Plotly python graphing library. Plotly. (n.d.). Retrieved October 24, 2021, from https://plotly.com/python/.

Removing stop words with NLTK in python. GeeksforGeeks. (2021, May 31). Retrieved October 24, 2021, from https://www.geeksforgeeks.org/removing-stop-words-nltk-python/.

Roshan, Ben. (2020) *Fake news classifier LSTM.* Kaggle. Retrieved December 4, 2021. https://www.kaggle.com/benroshan/fake-news-classifier-lstm

Seaborn. (n.d.). Retrieved December 06, 2021, from https://seaborn.pydata.org/.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22–36. https://doi.org/10.1145/3137597.3137600

Tensorflow. (n.d.). Retrieved November 22, 2021, from https://tensorflow.org/

Visualization with python. Matplotlib. (n.d.). Retrieved October 24, 2021, from https://matplotlib.org/.