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

Progress Report

Data Mining – CSPB4502

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

1. Introduction and Project Background

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.

Fake news and misinformation have been topics of growing conversation and interest over the past few years, and the field of study on this topic has been growing as well. In their paper “Fake News Detection Using Machine Learning Approaches”, authors Khanam, Alwasel, Safari, and Rashid discuss advanced methods of building AI machine learning models that can effectively differentiate fake news from real news. Shu, Silva, Wang, Tang, and Liu’s “Fake News Detection On Social Media: A Data Mining Perspective” takes a similar view to the fake news detection. Both of these works focus on the problem from the perspective of computer science, which is decidedly different from the objective of this project.

We will be using similar methodologies to classify our data sets, but from there, we will be analyzing the real and fake articles for key similarities and differences on the body text of the articles alone. We are not primarily focusing on the AI driven classification of real vs fake news, but analyzing the articles once the classification has been made for differences that could potentially be caught by the average news reader.

2. Data Preprocessing

2.1 Pre-processing

To start our project, our group gathered multiple data sets from across a variety of sources (See Section 2. Datasets). The first thing that we need to do is to combine each of these data sets into one Master set. This involves getting each of these data sets ready to be integrated with one another. For example, the dataset from Common Crawl titled “News dataset containing news from news sites all around the world.” was a dataset containing around 700,000 news articles in English from various news sites hosted around the world. The problem was that each of these data points was provided as its own, separate CSV file. So, to get all of this data into one place, a custom python program was written by the team to gather all of the relevant data from each of these individual files and write them to one large CSV file. A similar process is needed for each of the data sets to prepare them to be all converted into one large data set to be evaluated.

2.2 Data Cleaning

The next step in preparing the individual data sets is cleaning them. This includes multiple steps, such as removing duplicates and unnecessary data . We will be identifying and removing the words from every news article that do not have a significant meaning. The list of words can be found by using “stop words” from the Natural Language Tool Kit (nltk) library (See Section 5.8 : 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 stop words are “the”, “and”, “an”, “because”, and “of”.

2.3 Data Classification

Many articles from the datasets we are evaluating are pre-classified as fake news and real news, but not all of them. The goal of our project is to be able to use a Python AI-Model to classify all articles that have not already been classified as real or fake.

We will use our pre-classified data sources as supervised training data.

2.4 Data Integration

In this step, we will combine all the datasets labeled as fake and true news articles to obtain one large dataset. We will host the dataset in a Postgresql database which will be hosted on Heroku. We decided to use a cloud option so that the dataset would not need to be individually downloaded on each member’s system.

Having a single repository would enable us to make any changes or updates to the data once without the need to individually update all members’ systems. Hosting on Heroku also allows 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.

In order 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 of our 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.

2.5 Data Transformation

After combining all the datasets and obtaining the lists of words of each news article, the strings will be converted to tokens (numbers) for easy data manipulation. This is known as tokenization.

2.6 Data Reduction

Lastly, the data will be reduced by eliminating repeated words and keeping only the unique words contained in each news article.

3. Datasets

Title	**Source**
A Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles	Harvard Dataverse
News dataset containing news from news sites all around the world.	Common Crawl
Dataset consisting of fake and real news articles.	Kaggle: Clément Bisaillon
Fake news detection dataset consisting of train and test data for fake news model development.	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

4. 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.

4.1 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.

4.2 Numpy

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

4.3 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.

4.4 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.

4.5 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*) We will be using it for deep learning in our AI model to classify fake vs. real news.

4.6 Natural Language Toolkit for Python

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

4.7 Tensorflow

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

5. Evaluation

Confusion Matrices will be used to view the performance of our classification model. We will use our set of test data, for which we know the true values, to correct against our model results. Assembling the Confusion Matrix will allow us to quantifiably evaluate the results of our model. We will use the metrics of Accuracy and Precision, to measure exactness.

We will use a Recurrent Neural Network (RNN) to train our model of tokenized word vectors by utilizing the added temporal or sequential aspect that RNN allows us. We will also augment our RNN with Long-Short Term Memory (LSTM) to be able to establish long term dependencies between entailments and presuppositions in utterances. Using these modified Neural Networks will allow us to better process and correlate related semantic and syntactic features.

6. Milestones Completed

Task	**Completion**
Data Preprocessing	November 28, 2021

Data preprocessing steps:

The datasets have been downloaded into a local drive. The csv files were manually manipulated in Excel to make sure they all contained the information needed to work in

the training model. The datasets were only left with the columns: 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. The Pandas library was imported to help in the process of extracting, transforming, and loading the data integration of the files.

Before working on the data integration, a categorical fix step had to be performed to make sure that all the datasets displayed only three columns as several of the other columns were initially included in my extraction. We also had to transform the labels into 0s and 1s as some of the articles were labeled as 'FAKE' or 'REAL'.

The label column intends to answer the question "is it fake?", so it was necessary that all the news had a binary representation of 0 meaning the article is true and 1 that is fake.

After labeling correctly all the articles, the data integration was the next step. The following three datasets were concatenated as master dataset to later be used for testing purposes:

Kaggle: Samrat Sinha as File1

The Fake News Dataset as File2

Kaggle: Paul Larmuseau and Deeachain as File3

The datasets (True and Fake) from Kaggle: Clément Bisaillon were chosen to use as training data for this project. Both files have been concatenated as train data.

The master and train dataset had to be modified by adding a new column labeled 'original'. This column contains the title and text information combined, as this one will be used in the data analysis step.

7. Milestones To-Do

Task	**Completion**
Data Transformation and Reduction	December 2, 2021
Train Model	December 6, 2021
Accuracy Results	December 8, 2021
Final Project Report	December 10, 2021
Project Code Submission	December 10, 2021
Project Presentation	December 10, 2021

The Zenodo: Radu Prodan, Prateek Agrawal, Pawan Verma dataset that will be represented as File4 in the Python module, it is written in binary code and because of this it has not been incorporated to the master dataset yet. This dataset is expected to be decoded by December 1, 2021.

8. Current Findings

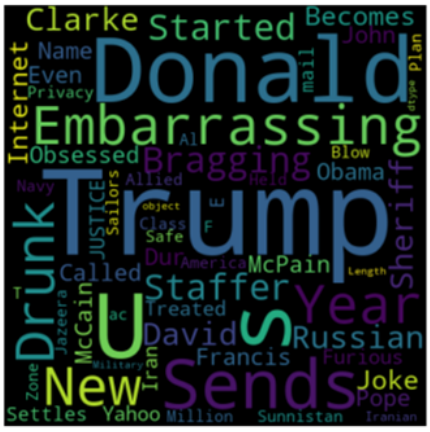
In the initial steps of the data processing, a word cloud has been created of three groups: 1) text alone, 2) titles alone, and 3) text and titles. This was created with the idea of investigating the importance of

including both the titles and the text versus just one alone. This allows us to better understand the nuisances involved in our machine learning algorithm.

The following word clouds are based on the fake dataset from [Kaggle: Clément Bisaillon]

Article Titles

Text Content



Article Titles and Text Content

9. Sources

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.

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

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/.

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

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

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

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