FAKE NEWS DETECTION

CASE STUDY

Qasim Ali

Logini Maheswaran

Rich Zhang

Executive Summary

Up until birth of these social media such as twitter, Facebook and YouTube, people consumed news mostly via printed newspaper, TV and radio. Through these mediums, people did not have to question the source and validity of the news. We are in a totally different era with news in the current social media world. Spreading fake news has become easy and it is a serious issue. A few facts on fake news in the United States and United Kingdom:

- Propagation of fake news has had a non-negligible influence of 2016 US presidential elections
- 62% of US citizens get their news for social medias
- Fake news had more share on Facebook than mainstream news
- Fake news has also been used in order to influence the referendum in the UK for the “Brexit”

More than ever year 2020 seems to be on another level when it comes to spreading fakes news and people being divided on the opposite spectrum when it comes to politics. Fake news has the power to destroy the democracy of a country by diving people. Because of all the things happening currently, we thought doing the case studies in fake news detection will be interesting and relevant. We contributed to this project by following the six-sigma management technique to autonomously detect the fake news.

Figure 1: Workflow during the case study.

For this case study, we found data that contains two types of articles: fake and real political and world news. The real dataset was collected from real-world sources like Reuters.com. The fake news articles were collected from unreliable websites that were flagged by PolitiFact and Wikipedia.

We applied a few Natural Language Processing – Text Classification models such as K-Neighbors, Decision Tree, Random Forest, Logistic Regression, Naïve Bayes and Support Vector Classifier. Decision tree, Logistic Regression and Support Vector Classifier had promising results with all above 98% accuracy. K-Neighbors didn't achieve desired accuracy.

We are hoping these machine learning techniques and responsibility from the social media companies restricting fake news will restore some normalcy in consuming true news.

Problem Description

Communication through social media about any subject does not properly identify the originality of a statement or story. Political leaders and large media channels use social median platforms to spread news in the public by tarnishing statements of their opponents and sometimes show misleading contents. This attract audience on their platform and make it easier to achieve short-term goals or take benefit from ongoing situations. Such prospect is augmented in past few years with the emergence of user-friendly social media apps and has become a dangerous prospect for online users. Therefore, it is important to identify and measure the originality of posted text on social media.

Figure 2: Detecting Fake News

Machine learning algorithms plays a vital role in identifying fake and true contents. In this project, we are interested to build a learning model that can predict opinion spams and fake news about any popular entity. We will gather data from online resources, preprocess the data to clean it, apply some natural language processing techniques to identify the words, train a model to learn and associate those words with target variable and test the model on a given data set.

Data Description

We worked on true and fake news dataset that is collected and compiled by ISOT research lab, University of Victoria, Victoria, BC, CA. The data is available to public for research and exploration purposes at the following URL:

https://www.uvic.ca/engineering/ece/isot/datasets/index.php

The dataset consists of several thousands of fake news and true news articles. The news that is collected from unreliable resources is called as fake news dataset, whereas the news collected from authentic websites is considered as true news dataset. Most of the true data was collected during the years 2016 and 2017 from reuters.com by exploring few aspects of the given datasets, we found that the data is relevant to world and political news. Fake news dataset is comprised of news (39%), politics (29%) and others (32%) whereas true news data set is news (47%) and politics (53%). We also look at the uniqueness of the articles and found that 97% of the fake articles and 100% of the true articles are unique.

To understand the true and fake datasets in detail, we applied some feature extraction techniques and found some features that explains total number of articles, uniqueness in the articles, number of characters, words and numbers, alphabets, lower- and upper-case letters.

Feature Extraction	**True Data set**	**Fake Data set**
Number of Articles	21417 News: 10145 (47%) Politics: 11272 (53%)	23481 News: 9050 (39%) Politics: 6841 (29%) Others: 7590 (32%)
Unique Articles	100%	97%
# of Characters	Max: 29781 Average: 2384	Max: 51794 Average: 2547
# of Words	Max: 5175 Average: 394	Max: 8436 Average: 435
# of Numbers	Max: 72 Average: 2.35	Max: 118 Average: 1.7
# of Non-Alphabets	Max: 798 Average: 59	Max: 7295 Average: 59
# of Lowercase	Max: 4394 Average: 308	Max: 6601 Average:340
# of Uppercase	Max: 214 Average: 6.72	Max: 309 Average: 8.92

Data preparation details

Main data preparation tools from the packages and libraries of the python made data ready for the machine learning models. Natural Language Tool Kit (NLTK) is a comprehensive package that includes libraries for tokenization (word_tokenize), Stemming (PorterStemmer), lemmatization (WordNetLemmatizer) and stopwords (Stopwords). These libraries along with plotting and data handling libraries were used to remove punctuation marks, numbers, URLs, HTML tags, Emojis and stop words and visualizing the dataset.

Lemmatization and stemming of words have different purposes yet they both are equally applicable in our case. Using lemmatization, the words are converted to their dictionary form while stemming follows an algorithm that is faster and provide vocal forms of the words. Since the datasets are comparatively small and there is enough time to run the simulations, lemmatization as well as stemming was performed. In addition, few classification machine learning algorithms (logistic regression and multinomial naïve Bayes) were executed to test the accuracy of the model. The difference in the accuracy for both data preparation techniques lied under one percentile to each other.

The data preparation leads us to separate out the key words in each dataset. Next, the data was visualized by using world-cloud (WordCloud) to find prominent words in the true and fake news datasets as shown in Figure 3a and Figure 3b.

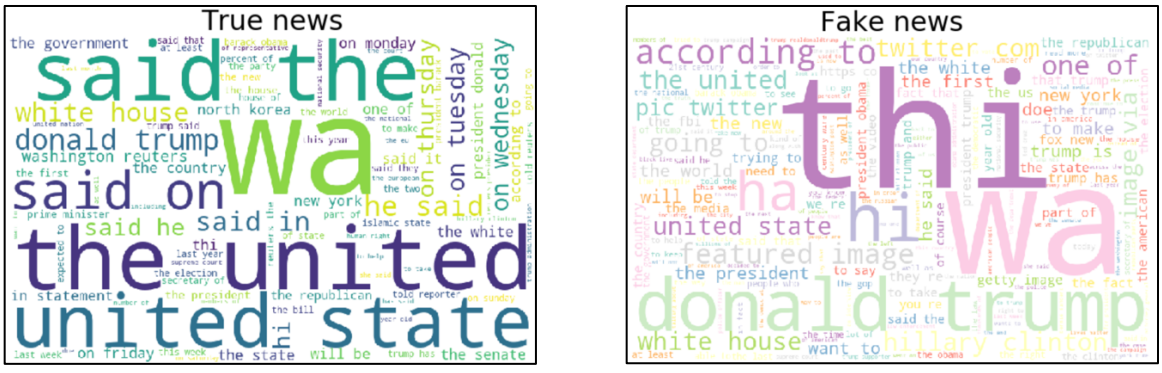


Figure 3: Word-cloud for the true and fake news datasets after data preparation.

Data Analysis Solution

Fake news detection is essentially a text classification problem. We applied machine learning classification models in Python Scikit Learn library for model training and testing. The preprocessed text dataset was randomly split into training and testing dataset in a 75:25 ratio. The training dataset was used for training the model and the testing dataset was used for model validation. We ran the dataset through multiple models, which allowed us to compare and evaluate the individual model performance and pick the best model for this application. The models we applied include K-Nearest Neighbors, Decision Tree, Random Forest, Logistic Regression, Naïve Bayes and Support Vector Classifier.

The model performances were evaluated on the testing dataset by the model metrics, Accuracy, Precision and Recall. Accuracy shows the overall model performance. Precision and Recall are also important metrics for classification problems, in which high Precision score means less real news samples being labeled as fake news, and high Recall score indicates that less fake news samples being labeled as real news. In the Test Dataset section of the following table, it shows that three metrics of all the models are over 90% except that K-Nearest Neighbors has low Accuracy and Precision scores. These results indicate that most of our models are capable of detecting fake news articles accurately in the given dataset.

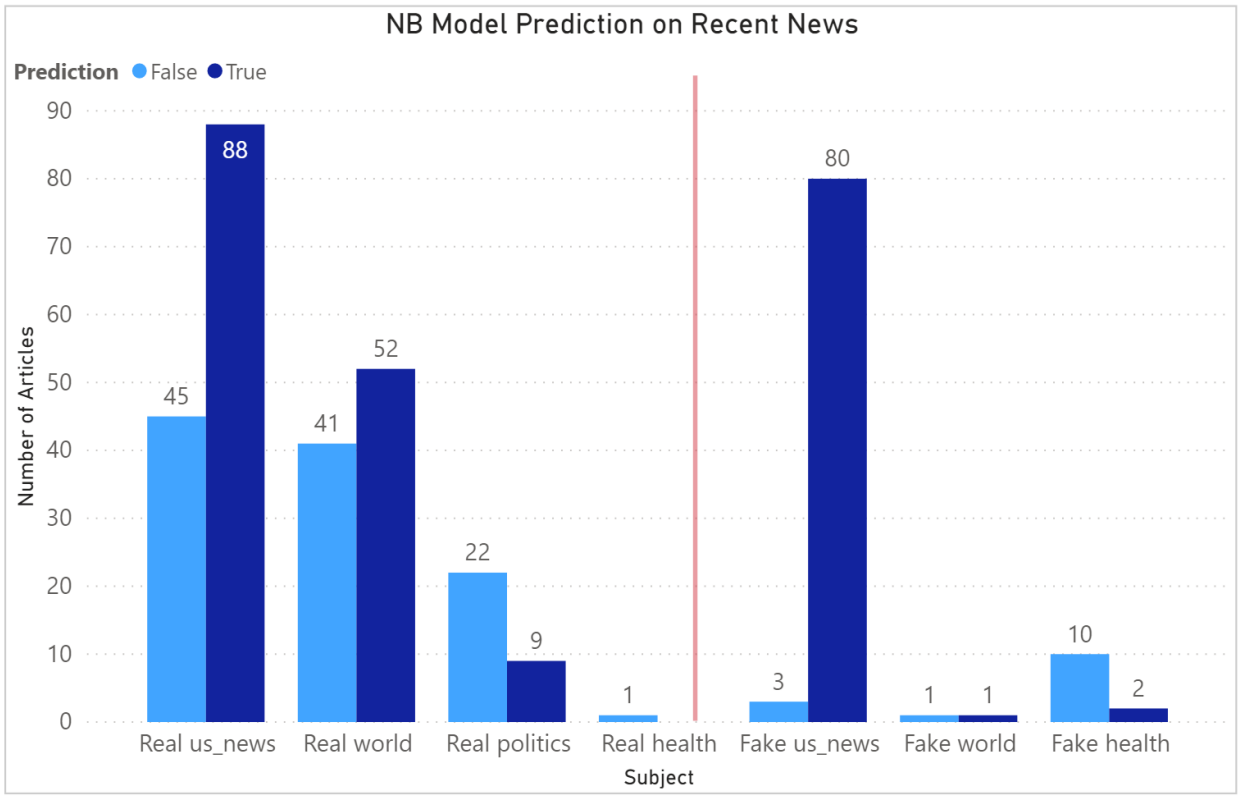
Table 1

Model Name	Testing Dataset			Recent News		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
K-Neighbors	69.1	63.4	97.5	29.8	28.0	1.0
Decision Tree	99.6	99.6	99.7	33.8	29.2	1.0
Random Forest	96.2	97.9	94.8	66.5	44.0	82.5
Logistic Regression	98.6	99.0	98.3	58.3	39.4	97.9
Naive Bayes	93.0	92.9	94.0	65.0	43.0	85.6
SVC	99.4	99.6	99.3	54.9	37.6	99.0

However, this fake news dataset was collected before March 2018. It is of our interest to see how the model would performance on recent news articles (after 2019). We collected a recent news dataset consisting of 258 real news articles (web scraped from reliable news website) and 97 fake news articles (web scraped from fake news websites tagged by PolitiFact). Even though this is a small dataset but it

would give us an idea of our model performance on the recent news. The evaluation results are shown in the Recent News section of Table 3. It can be seen that all the models have lower accuracy on the recent dataset. The K-Nearest Neighbors and Decision Tree have the worst performances due to very low Precision scores despite perfect Recall scores. The Random Forest and SVC have the best overall Accuracy with a trade-off of lower Recall for higher Precision. Taking the results of Naive Bayes model as an example for illustration, the following figure shows that the model is able to detect fake news accurately but also labeled a big portion of real news as fake news. This suggests that given the fact that our models were trained on an old dataset, they cannot be applied to the recent data directly. Likely, this is because our models use Bag of Words - TF method for feature extraction, which weighs the importance of words based on the number of their appearances in the articles. This could limit our model's ability to predict recent news since recent news would focus on different topics and use different set of vocabularies. To make the model useful for real world application, one solution is to re-train the model frequently with new data to keep the model up to date. Another approach could be to use Word embedding for feature extraction instead of Bag-of-Words. Unlike Bag-of-Words, Word embedding method like word2vec or GloVe represents words in vector space and preserves contexts and relationships of words. Therefore, Word embedding method could potentially overcome the inherent limitation of Bag-of-Words method we are facing in this application.

Figure 4



Conclusions

Six machine learning models were trained and tested on the fake news dataset. Except K-Nearest Neighbors, all the models have a score over 90% on the three metrics (Accuracy, Precision and Recall). Especially, the metrics scores of the Decision Tree and SVC models are all over 99%. These results indicate that our models are capable of accurately detecting fake news in the given dataset.

However, the performance of the models drastically deteriorates when tested against recent news data. This could be attributed to the inherent limitation of the classic Bag-of-Words TF method the models used for feature extraction. The model would need to be re-trained frequently with new news data for real world application.

Word embedding method represents words in vector space and preserves contexts and relationships of words. Therefore, feature extraction using Word embedding method could likely overcome the limitation of Bag-of-Words method we are facing in this application. This could be an interesting direction for future work.