Machine Learning Algorithms for Fake or Real News Detection

Ziqian Xu, Yinuo Hu, Mingli Zhang, Xinran Ming

1 Introduction

Fake news is news or stories created to deliberately misinform or deceive readers. Usually, these stories are created to either influence people's views, push a political agenda or cause confusion and can often be a profitable business for online publishers. They might also be satire/parody sources such as *The Onion* that was created as a form of entertainment.(Brodie, 2018)

False information can deceive people by looking like trusted websites or using similar names and web addresses to reputable news organisations.(University of Michigan Library, 2020) Thus, it is important for us be able to identify the truthfulness, accuracy, and authenticity of the news and associated information.

Previous studies have attempted to use machine learning algorithms to distinguish fake news from real ones. Researchers have used Logistic Regressions, Naive Bayes algorithms, Support Vector Machines, and Decision Trees to classify news (Katsaros et al., 2019; Granik et al., 2017). Different types of artificial neural networks including Long Short-Term Memory (LSTM) Recurrent Neural Networks, Convolutional Neural Networks, and Adversarial Neural Networks were also widely used in fake information detection tasks (Bahad et al., 2019; Wang et al., 2018; Yang et al., 2018).

This variety of available algorithms for information identification has lead to a new question: what is the "best" way of fake news identification? Our project was designed to address this need and to explore the effectiveness of different machine learning methods on news classification. Taken together, we aim to compare the performance of Logistic Regression, Naive Bayes, Support Vector Machine, and LSTM neural network in fake news classification, via evaluating their accuracy and the confusion matrices corespondent to each algorithm.

2 Data and Preprocessing

2.1 Data

We obtained two datasets, one with all fake news (N = 23481) and one with all real news (N = 21417), from https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset. They each contains 4 variables, including the title of the news, the text content of the news, the subject of the news, and the date for which the news are posted.

We used python and its built-in libraries in this project. We used pandas, numpy, matplotlib, seaborn, nltk, and wordcloud for simple text data manipulations and visualizations. After that, we used scikit-learn and keras to train machine learning algorithms for news classification.

2.2 Preprocessing

2.2.1 Merging Data

We first added an additional indicator variable indicating fake (1) or real (0) in the both the real news dataset and the fake news dataset. Then, we merged the two datasets into a complete dataset containing all the news (N = 44898).

2.2.2 Text Cleaning

We used the text contents of the news to classify whether they are real or fake. To do so, we first conducted the following steps to clean up text content of each news article:

- Remove external links in the text section, including tweet quotes, urls, slashes. The criteria for exclusion include but is not limited to "@", "https://", ".com", ".org", "/".
- Remove non-informative contents in the text section, including punctuation, numbers, stop words, and non-alphabetic characters.
- Remove identity indicators, especially names of authentic news agencies. Such as "Rueter", "The New York Times", "The Washington Post".
- Used stemmer to identify the same words under different tense or plural forms. For instance, treat "drink" and "drinking" or "door" and "doors" as the same word.

Following these cleaning criteria, the example "WASHINGTON (Reuters) - The U.S. House of Representatives on Thursday approved an $81 billion bill to help widespread recovery efforts from hurricanes and wildfires this year." would be "the us hous repres thursday approv billion bill help

widespread recoveri effort hurrican wildfir year" after text cleaning.

3 Exploratory Data Analysis

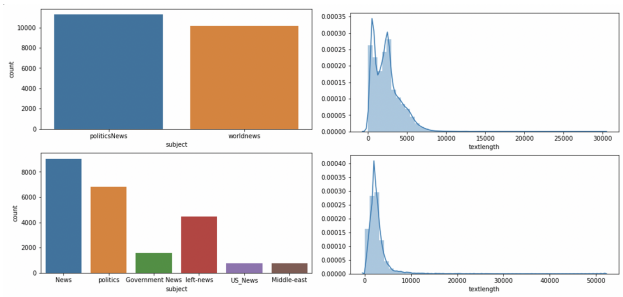

Figure 1. Distributions of news types and lengths

We explored the distributions of key news text features as preliminary analyses prior to building machine learning models. While real news only contained subjects of politics or world news, fake news evolved around six categories of subjects, including news, politics, government news, left-news, US news, and Middle East news. The distributions are shown in the Figure 1. Real and fake news also had different distributions of content text lengths also demonstrated in Figure 1: while real news had two peaks in the length distribution, fake news only had one such peak. Therefore, text length may be a good indicator of whether a news article is fake or real.

Figure 2. Text composition of real Vs. fake news

We also looked at the text composition of real versus fake news. The word cloud in Figure 2 show which words appear most frequently in the real news dataset and in the fake news dataset respectively. For both real news and fake news, a lot of keywords are related to the president of U.S. and are about politics. The differences in word compositions between real news and fake news are not clear from visual inspections of the word clouds, which is why we need to build machine learning models to classify them. Figure 3 and 4 further show real news and fake news word frequency distributions concordant to the top 50 most frequent words extracted by the word clouds. The word frequency distributions used raw data instead of the cleaned up data because whole words make more sense than stemmed words.

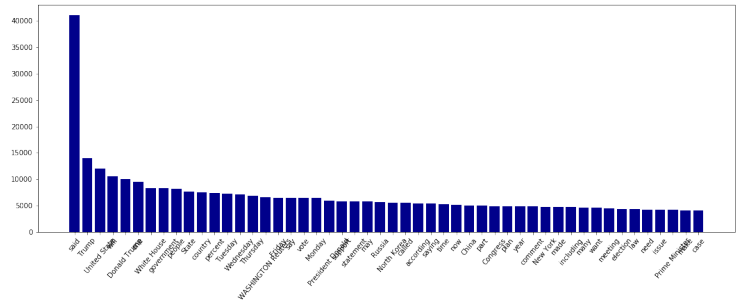

Figure 3. Real news word frequency

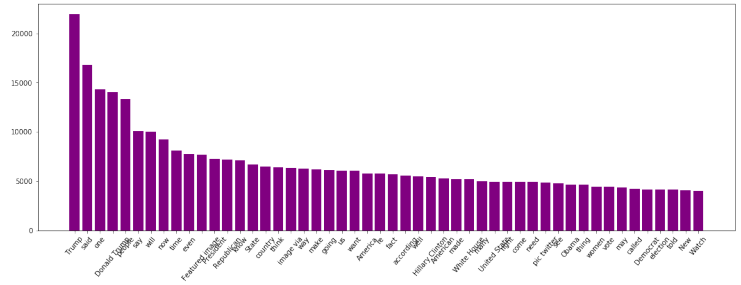

Figure 4. Fake news word frequency

4 Models

4.1 Text Encoding

We used different encoding methods for different models. For Logistic Regression, Naive Bayes, and Support Vector Machine, we used term frequency–inverse document frequency (TFIDF) to encode news text data into a sparse real-number matrix. For LSTM neural networks, we tokenized the news text corpus and used integer-encoding of each word as representation of data.

4.2 Training and Testing

To split training and testing set, we first drop columns with empty news article text. We then randomly chose 90% of data to be the training set (N = 39764) and the rest 10% of data to be the testing set (N = 4419).

4.3 Logistic Regression

Logistic Regression is widely used for binary classification problems. Logistic Regression uses the sigmoid function, $sigmoid(z) = \frac{1}{1+e^z}$, to convert linear relationships into binary cases using $y = \frac{1}{1+e^{-(b+w_1x_1+w_2x_2\ldots)}}$.

TFIDF transformation gave us 157664 features from the training set news text, and with these features, the Logistic Regression classifier predicts the testing set with an accuracy of 98.05%.

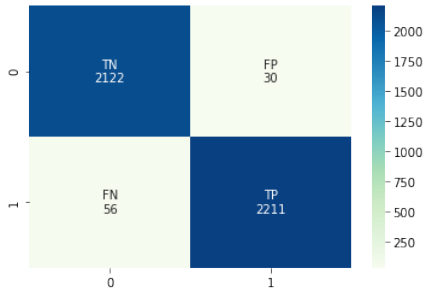

Figure 5. Confusion matrix for Logistic Regression

4.4 Naive Bayes

Bayes' Rule states that $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$. Naive Bayes applies Bayes's rule to a chain of features with the assumption that the probability of each feature is independent. There are different types of Naive Bayes classifiers that are based on different distributions such as Bernoulli, Gaussian, and Multinomial. Although the TFIDF encoding we used generates float-point data, Gaussian Naive Bayes classifier cannot be used in our study because it requires a non-sparse input matrix. Thus, we chose to use the Multinomial Native Bayes classifier, which would work with TDIDF data even though discrete encoding would be more ideal.

After training with 157664 features, the Multinomial Native Bayes classifier predicts the testing set with an accuracy of 93.32%.

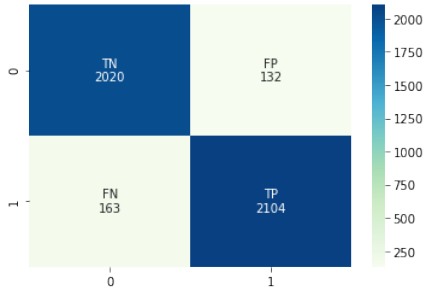

Figure 6. Confusion matrix for Naive Bayes

4.5 Support Vector Machine

Support Vector Machine (SVM) is another supervised machine learning algorithm often used for classification tasks. Unlike Logistic Regression, SVM uses the orthogonal distances from the decision boundary that separates different labels to the data points near the boundary to decided how to optimize the classification task.

After training with 157664 features, the SVM classifier predicts the testing set with an accuracy of 98.23%.

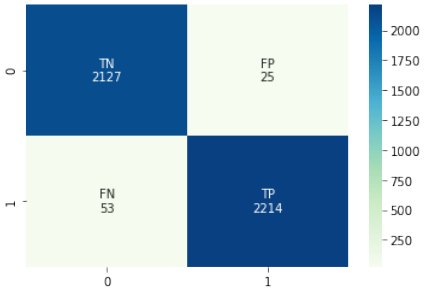

Figure 7. Confusion matrix for SVM

4.6 LSTM

We also used a bi-directional LSTM recurrent neural network suggested by Bahad et al. (2019) for text classification of real versus fake news. Bi-directional LSTM is a form of generative learning algorithm that connects hidden layers of two opposite directions as an attempt to connect what comes later with what comes before (Schuster & Paliwal, 1997). This is particularly useful in natural language processing such as text classification tasks because what comes later in a sentence may need to be combined with what comes earlier in a sentence to make sense.

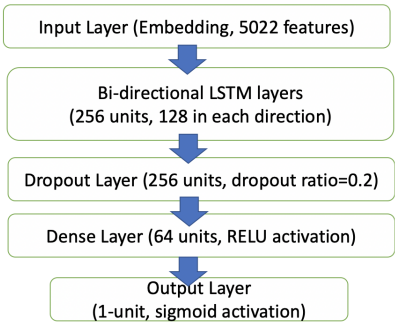

Figure 8. LSTM network structure

We used the above network structure with one iteration of the bi-directional LSTM layers followed by a dropout layer to prevent over-fitting. Two dense layers are used after the LSTM layers to generate the desired output. We trained this network with 5022 features in 39764 training samples over 5 epochs with a mini-batch size of 128.

This bi-directional LSTM neural network predicts the testing dataset with an accuracy of 98.73%.

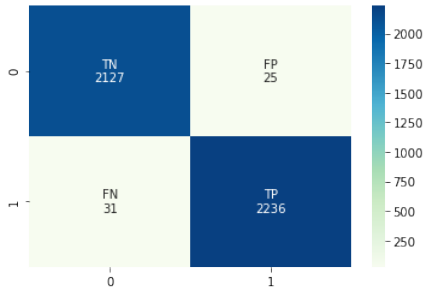

Figure 9. Confusion matrix for LSTM

5 Discussions

The following table shows the accuracies, true negatives (TNs), true positives (TPs), false negatives (FPs), and false positives (FPs) of the algorithms used in this study for classifying real vesus fake news.

model	accuracy	TN	TP	FN	FP
LR	98.05%	2122	2211	56	30
NB	93.32%	2020	2104	163	132
SVM	98.23%	2127	2214	53	25
LSTM	98.73%	2127	2236	31	25

Table 1. Algorithm Accuracy

Comparing the models above, the Naive Bayes classifier has the lowest accuracy while the LSTM network has the highest accuracy. However, although Naive Bayes has the lowest accuracy, the score is still above 90%. On the other side, while LSTM has the highest score, LSTM's accuracy is only marginally better than Logistic Regression and SVM. This being said, running more epochs during the training phase of LSTM may further increase the score. Admittedly though, LSTM is significantly more computationally expensive than the other algorithms, so the decision of which model to use may depend on both what accuracy level is needed and what computational power and time requirements are present.

Another thing to note is that although LSTM's accuracy is only slightly higher than the other machine learning models used in this project, we did train the LSTM model using much fewer features. The Logistic Regression, Support Vector Machine, and Naive Bayes had 157664 features to train on, whereas for the LSTM model, we only used 5022 features for training.

6 Conclusion

In conclusion, the machine learning algorithms we employed are relatively accurate in classifying fake news by text content analysis. There exists a difference in computational time among different methods. The difference in accuracy and running time are fully discussed in the discussion section. However, due to the lengthy nature of news contents, we find content analyzing extremely time-consuming, both in computation and training time. This might largely influence the efficiency of information identification. A solution to this problem might be an identification algorithm that focus on analyzing the title or summary of the news. Future work could be done on title-focused news classification, for the purpose of increasing computationally efficiency.

7 Acknowledgements

We thank the following kaggle kernels for guiding us through the natural language processing tasks:

- https://www.kaggle.com/parulpandey/getting-started-with-nlp-a-general-intro
- https://www.kaggle.com/nasirkhalid24/unsupervised-k-means-clustering-fake-news-87
- https://www.kaggle.com/muhammadshahzadkhan/is-it-real-news-nlp-lstm-acc-99-9Now-we-need-to-apply-padding-to-make-sure-that-all-the-sequences-have-same-length

References

[1] Ahmed, H., Traore, I., & Saad, S. (2018).Detecting opinion spams and fake news using text classification. *Security and Privacy, 1*(1), e9.

[2] Bahad, P., Saxena, P., & Kamal, R. (2019). Fake News Detection using Bi-directional LSTM-Recurrent Neural Network. *Procedia Computer Science, 165*, 74-82.

[3] Brodie, I. (2018). Pretend news, false news, fake news: The onion as put-on, prank, and legend. *Journal of American Folklore, 131*(522), 451-459.

[4] Granik, M., & Mesyura, V. (2017, May). Fake news detection using naive Bayes classifier. In *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)* (pp. 900-903). IEEE.

[5] Katsaros, D., Stavropoulos, G., & Papakostas, D. (2019, October). Which machine learning paradigm for fake news detection?. In *2019 IEEE/WIC/ACMInternational Conference on Web Intelligence (WI)* (pp. 383-387). IEEE.

[6] Research Guides. (2020, June 24). Retrieved November 19, 2020, from *https://guides.lib.umich.e du/fakenews.*

[7] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing, 45* (11), 2673-2681.

[8] Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., ... & Gao, J. (2018, July). Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery data mining* (pp. 849-857).

[9] Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., & Yu, P. S. (2018). TI-CNN: Convolutional neural networks for fake news detection. *arXiv preprint arXiv:1806.00749.*