Machine Learning Algorithms for Fake or Real News Detection

Ziqian Xu, Yinuo Hu, Mingli Zhang, Xinran Ming

1 Introduction

Fake news are gaining attention nowadays both due to the increasing use of internet and medias and due to the rise in political, social, and cultural awareness. News sources such as *The Onion* which intentionally publish fake information are also attracting attentions (Brodie, 2018). Thus, it has become more and more important for people to be able to identify the truthfulness of news.

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 (Bahad et al., 2019; Wang et al., 2018; Yang et al., 2018).

Because of the plethora of algorithms available to classify fake and real news, and because of the abundance of information of different qualities in our lives, the current study aims to compare the accuracies of Logistic Regression, Naive Bayes, and LSTM neural network [**add other algorithms to compare**] in classifying fake and real news.

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. The datasets each contains 4 columns representing 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.

2.2 Preprocessing

2.2.1 Merging Data

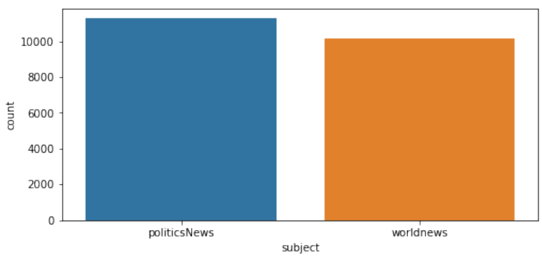
We first added an additional indicator variable indicating fake (1) or real (0) in the both the real news dataset and the fake news dataset. Then, we merged the two dataset into a complete dataset containing all the news (N = 44898).

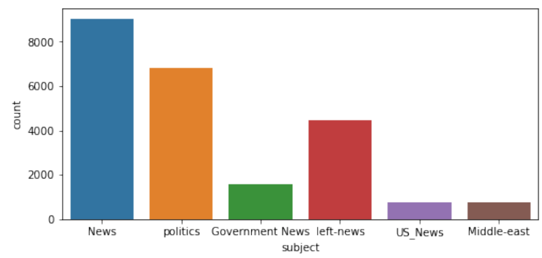
2.2.2 Text Cleaning

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:

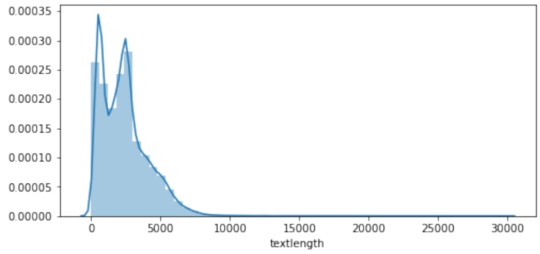
- remove text sections that are the news article tagging someone (e.g. "@aPerson").
- remove urls in the news articles (e.g. "https://twitter.com/").
- remove text sections that are not exactly urls but also contain slash (e.g. "pic.twitter.com/wiQSQNNzw0").
- remove "(Rueter)" and anything that comes before it. Real news in the datasets being used often begin with a location for the news and "Rueter" to indicate realness (e.g. "WASHINGTON(Rueter)...")
- remove punctuation
- remove numbers and non-alphabetic characters
- remove stop words, which are common words in a language that do not have specific semantic or syntactic significance (e.g. "the", "is")
- stem words into root forms to avoid different forms of the same word being treated as separated words.

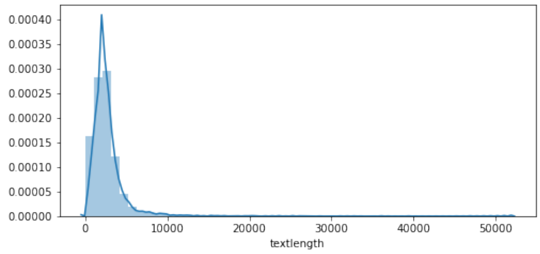
3 Exploratory Data Analysis





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 included six categories of subjects, including: news, politics, government news, left-news, US news, and Middle East news.





. Real and fake news also had different distributions of text length as shown in the right side figure: real news had two peaks while fake news only had one.

4 Models

4.1 Text Encoding

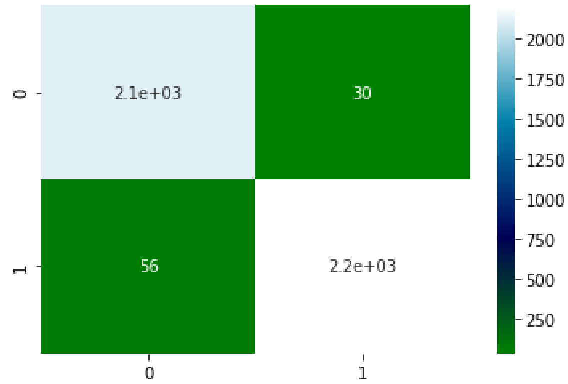
We used different encoding methods for different models. For Logistic Regression and Naive Bayes, we used term frequency–inverse document frequency (TFIDF) to encode news text data into a sparse real-number matrix. For LSTM neural networks, we tokenized the news text corpus and used integer-encoding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

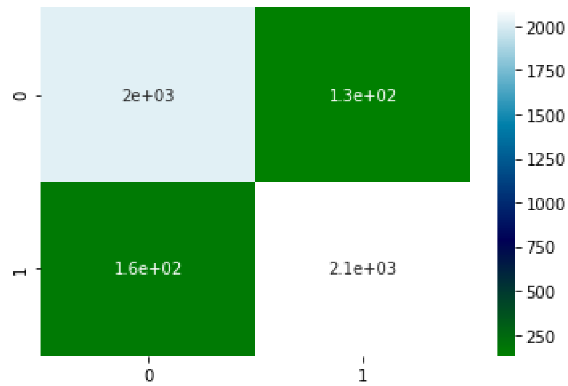
4.3 Logistic Regression

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%.



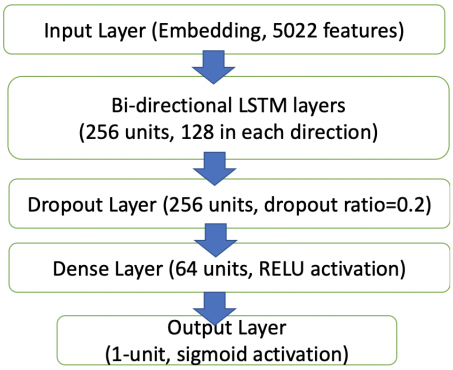
4.4 Naive Bayes

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

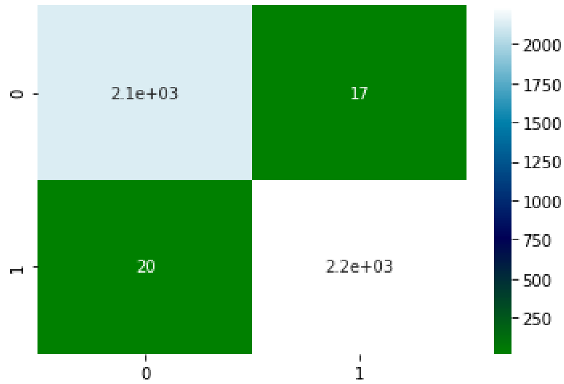


4.5 LSTM

We also used a bi-directional LSTM recurrent neural network suggested by Bahad et al. (2019) for text classification. Bi-directional LSTM is a form of generative learning algorithm that connects hidden layers of two 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 because what comes later in a sentence may need to be combined with what comes earlier in a sentence to make sense. We used the following network structure with one iteration of bi-directional LSTM followed by a dropout layer to prevent overfitting. Lastly, two dense layers are used to generate output. We trained this network with 5022 features in 39764 training samples over 5 epoches with a mini-batch size of 128.



This bi-directional LSTM neural network predicts the testing dataset with an accuracy of 99.16%.



5 Conclusions

5.1 Model Comparison

model	accuracy	TN	TP	FN	FP
LR	98.05%	2122	2211	56	30
NB	93.32%	2020	2104	163	132
LSTM	99.16%	2135	2247	20	17

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, running more epochs during the training phase may further increase the score. Admittedly, LSTM is significantly more computational expensive than the other algorithms, so the decision of which model to use depend on both what accuracy level is needed and what computational power and time requirements are present.

5.2 Future Directions

Machine learning algorithms are relatively accurate in predicting fake news from the news articles' text content. However, because news articles can be long, processing and categorizing article texts maybe inefficient in computation and training time. In the future, we may investigate whether instead of news article content, news title line can be used to predict their truthfulness.

6 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/muhammadshahzad khan/is-it-real-news-nlp-lstm-acc-99-9Now-we-need-to-apply-padding-to-make-sure-that-all-the-sequences-have-same-length

7 References

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

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

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.

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.

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

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

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).

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.