Fake News Detection: Comparing Classifiers

Tina Yuan
New York University
thy258@nyu.edu

Pin-Tsung Huang
New York University
pth254@nyu.edu

Abstract

Fake news has been around even before the Internet. Sometimes, it is hard to tell between true and fake news, which leads to further spreading of such misinformation. We would use articles that are tagged as either "real" or "fake" and perform classification on the data. We would like to test different classifiers and compare their performance and efficiency in detecting fake news. We are comparing performance of four classifier models: Multinomial Naive Bayes, Support Vector Machines, Long Short-Term Memory and Random Forests. Surprisingly, all four models trained on the dataset achieved an accuracy above 0.97. The experiment result suggests that the pre-trained model is unable to accurately identify fake news in the new test data. To further elaborate on our studies, we decided to explore whether or not fake news classifier models trained on one data set can be used upon inference on another data set.

1 Introduction

Fake news has been around even before the Internet. The situation has worsened since the early days of the Internet: the connected nature of the Internet makes information being shared very rapidly regardless of its truthfulness. Leading to distrust, misinformation has become a major threat to democracy. Sometimes, it is hard to tell between true and fake news, which leads to further spreading of such misinformation. We would use articles that are tagged as either "real" or "fake" and perform classification on the data. We would like to test different classifiers and compare their performance and efficiency in detecting fake news.

2 Models

We are comparing performance of four classifier models: Multinomial Naive Bayes, Support Vector Machines, Long Short-Term Memory and Random Forests.

2.1 Multinomial Naive Bayes

Multinomial Naive Bayes (Kibriya et al., 2004) is a very simple, efficient yet effective classifier when it comes to natural language processing tasks. Naive Bayes makes a strong assumption: the input features are conditionally independent given the label. While it is mostly not true for inputs of a classification task, its performance is usually surprisingly good.

The major upsides for Multinomial Naive Bayes are simplicity, speed and ability to expand. Based on conditional probabilities, it is easier to understand and implement, and the result is more interpretable. Multinomial Naive Bayes model is fast: short training time and short inference time. This enables real-time applications on resource-restrained platforms such as embedded devices. It is also expandable since we can customized, task-specific features instead of plain words or n-grams.

A clear disadvantage is that Naive Bayes assumes all features are independent, which is not true in most scenarios.

2.2 Support Vector Machine

Support Vector Machine (Noble, 2006), or SVM, creates a hyperplane to separate data into exactly two classes. This is suitable for our purpose: separating fake and true news.

The major advantage of Support Vector Machines is the ability to take relations among features into account. Its performance is generally better than that in Multinomial Naive Bayes, and it is more effective in higher dimensional spaces.

The downside Support Vector Machines is we need to determine a proper kernel function, which is not a trivial task. Meanwhile, unlike Multinomial Naive Bayes, it does not have probabilistic interpretations.

2.3 Long Short-Term Memory

Long Short-Term Memory (Hochreiter and Schmidhuber, 1997), or LSTM, is a special kind of recurrent neural network (RNN) architecture used in deep learning. LSTM has feedback connections.

The major advantages of Long Short-Term Memory include the ability to take contexts into consideration due to its memory. Also, as a neural network model, it can process continuous streams of data. At the same time, it avoids the major downsides of RNNs: vanishing and exploding gradients.

The disadvantage of Long Short-Term Memory models is that it is prone to overfitting, and it is very resource-demanding.

2.4 Random Forests

Random Forests (Pal, 2005) are an ensemble learning method for classification. It operates by constructing given numbers of decision trees when it is being trained. The output of the random forest is the class selected by most trees when it comes to classification tasks. The model should be resilient to overfitting since the forests are randomly restricted to be sensitive to only selected feature dimensions.

The major advantages include resilience to overfitting, better performance compared to decision trees and less configuration effort needed over broad range of data.

It also has some disadvantages. There are much more parameters to tune, which adds complexity to the model. It is also more resource-demanding since hundreds of trees need to be calculate and they could consume a lot of RAM. And it is less interpretable than a single decision tree.

3 Methodology

3.1 Datasets And Pre-processing

We use data from a dataset on Kaggle–Fake and real news dataset: Classifying the news–with 44,898 entries (Bisaillon, 2019). There are four columns in the dataset: 'title', 'text', 'subject' and 'date'. We use 'title' and 'text', merging them into a single column.

Since we are doing plain comparison among the four models, we exclude unnecessary columns from the three datasets i.e. we are only keeping the main text and the fake/real label. A dedicated script is used to remove columns, unify the representation of fake and real, concatenate the three datasets and remove null entries. We also convert texts to all-lowercase and do text processing to remove:

- URLs
- Network protocols (e.g. https://)
- Excessive whitespaces and newlines
- Stopwords

4 Results

4.1 Model performance and efficiency

Surprisingly, all four models trained on the dataset achieved an accuracy above 0.97. The simplest model, the multinomial naïve bayes, provides similarly high model accuracy when ran on the data set compared to the random forest classifier and to the SVM and LSTM models. As shown in table **??**, the model that produced the highest accuracy is the LSTM network. Yet, this model also took the longest time to train, running at one hour of training time for 10 epochs. The second most accurate model is the Support Vector Machine. It has a much shorter training time of approximately 2 minutes. The training results and model performance suggest that the improvement in model accuracy as achieved by the LSTM network is not worth the 30 times increase in training time as compared to the SVM. Both the random forest classifier and the multinomial naïve bayes model provide real-time results during inference time. Therefore, the random forest classifier would be a good model to deploy for real-time systems as it has a slightly higher accuracy compared to the multinomial naïve bayes while still maintaining a low inference time.

Figure 1 suggests that all models perform similarly at identifying true fake news and true real news.

This result is expected as all models have high accuracy. The multinomial naïve bayes model, with the highest false real news rate, would be the least ideal model to deploy for fake news detection as the high false real rate may end up misguiding users and feed on spreading misinformation.

4.2 Inference on another data set

To further elaborate on our studies, we decided to explore whether or not fake news classifier models trained on one data set can be used upon inference on another data set. The experiment was

Model Type	**Accuracy**	**Training Time**	**Inference Time**
Multinomial Naïve Bayes	0.979	0.21 sec	0.12 sec
Support Vector Machine	0.99	1 min 58 sec	20 sec
Random Forest Classifier	0.984	31.8 sec	0.96 sec
LSTM	0.99	1 h 2 mins	34.9 sec

Table 1: Model Performance and Efficiency.

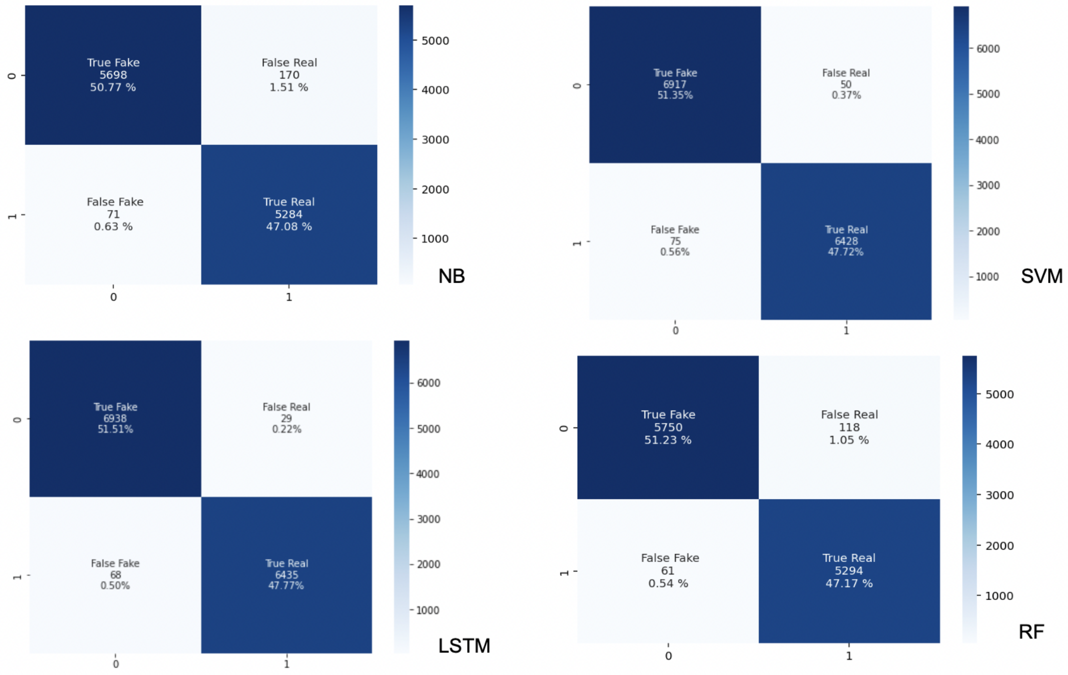

Figure 1: Confusion matrices of the 4 models ran on the original data set. Multinomial Naive Bayes (top left), Support Vector Machine (top right), LSTM (bottom left), and Random Forest Classifier (bottom right).

Model Type	Accuracy
Multinomial Naïve Bayes	0.57
LSTM	0.58

Table 2: Model Performance when ran inference on politifact dataset.

conducted using the LSTM and multinomial naïve bayes models that were trained on the data set mentioned in Methodology and were later tested on the data set below. A new data set is loaded from Kaggle that includes news articles curated by politifact.org (Ganesh, 2021).Politifact.org has a team of experts that determine whether or not an article is fake news or not. The same data preprocessing techniques were applied to the data to yield the final data matrix that included just the text and label columns. As shown in figure 2, both models achieve low accuracy when running inference on test data. The experiment result suggests that the pre-trained model is unable to accurately identify fake news in the new test data. The low accuracy may be caused by the difference in the data distribution between the two data sets. The first data set has an average token length of 242 tokens per text example while the politifact data set has an average token length of 11. Since the models were trained on the longer data set, the models may consider number of tokens as a determining feature and mark the majority of texts with fewer tokens with one label. This would lead to poor performance on the test data as the test data has fewer tokens than the training data in general.

4.3 Training on both data sets

Since training on one dataset and inferring on another didn't produce great results, we decided to try training a model on both datasets and then testing on held out data from the hybrid data set. The experiment was conducted using the multinomial naïve bayes model and the accuracy achieved by this model is 0.88. Although the combined data set provided much better model performance compared to the previous approach, it still produced subpar results compared to training on a single data set. The experiment suggests that when training on smaller data sets, large variance in the data distribution can still lead to poor model performance.

5 Discussion

The results of our first experiment (comparison of model performance and efficiency between 4 different classifiers) suggest that although the LSTM network is able to provide the highest accuracy, its long training and inference time makes it a far from ideal candidate when deploying a fake news detector. The SVM model is a good choice for deployments that want to achieve high accuracy and relatively short training and inference time. For real-time systems that require low inference time, the random forest classifier is a good candidate as it provides good accuracy and short inference time. It also performs better than the multinomial naïve bayes model where a much higher ratio of false real articles were identified. Future work can be conducted on exploring model performance when training on a larger data set that includes more varied text data and then running inference on another data set that may be completely different. Future work may also perform data clustering on the train and test data to determine which features affect model performance the most.

References

Clement Bisaillon. 2019. Fake and real news dataset: Classifying the news.

Shiv Kumar Ganesh. 2021. Politifact factcheck data.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2004. Multinomial naive bayes for text categorization revisited. pages 488–499.

William S Noble. 2006. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567.

Mahesh Pal. 2005. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222.