COVID-19 FAKE NEWS CLASSIFICATION

TECHNICAL REPORT

Anonymous Author
Department of Computer Science
University of Durham

May 23, 2021

1 Introduction

The news plays an important role in society as it informs the public on both local and global events that may affect them. One event in particular that has seen an extensive amount of coverage over the past few years is the COVID-19 pandemic. During this time, there has been an abundance of fake news and misinformation which has often led to widespread confusion and panic. Organisations such as The United Nations and the World Health Organization have labelled this spread of fake news and misinformation as an infodemic, and have acknowledged the damaging effects that COVID-19 related fake news has had on society. To tackle this on-going issue, it is essential that fake news and misinformation is able to be quickly detected and prevented from being broadcast to vulnerable individuals.

In this paper, we approach the task of detecting fake COVID-19 news by combining the fields of natural language processing and machine learning to produce a number of models that are able to correctly classify COVID-19 related articles as real or fake, using primarily the article's headline. We focus exclusively on supervised learning methods, and produce a total of six models, four of them being traditional machine learning models, and the other two being deep learning models. A number of different feature extraction techniques are utilised, and the performance of each model is captured, allowing them to be evaluated and compared against each other. We focus directly on the accuracy achieved by each model on a fixed test set to determine the best performing model. We also analyse some of the methods used by the models to classify fake news, which provides an interesting insight into the kind of language that is associated with both real and fake COVID-19 article headlines.

2 Dataset

In order to create our models, we required a large amount of labelled data that could be used for training. Producing a self-made dataset containing real and fake COVID-19 articles was heavily considered, however, we ultimately decided against this as it proved to be extremely difficult and time consuming to correctly classify whether a given article posted online contained real or fake information. Perhaps one of the more challenging aspects of this task was acquiring a suitable dataset containing this kind of labelled data. We ended up using the COVID Fake News Dataset provided by Banik [2020]. To our knowledge, this is the first time this dataset has been used in an academic setting. The dataset contains over 10,000 COVID-19 article headlines posted online throughout 2020, each appropriately labelled with either a 0, indicating it was from a fake news article, or a 1, indicating it was from a real news article.

2.1 Implementation Tools

We created our models on Python, which meant we could use many of the existing libraries to aid our development. Almost all of the natural language processing was done using the Natural Language Toolkit, Bird et al. [2009]. For the traditional machine learning models, both feature extraction and the classifiers were implemented using the sklearn library, Pedregosa et al. [2011]. The deep learning models were built using Keras in the tensorflow library, Abadi et al. [2015], and utilised the pre-trained 300-dimensional GloVe word embeddings provided by the Stanford NLP team, Pennington et al. [2014].

(a) Real news (b) Fake news

Figure 1: The most common words contained in the dataset after preprocessing

3 Methodology

In order to correctly classify COVID-19 articles as real or fake, a specific series of events was undertaken. This included first preprocessing the data provided in the dataset, then extracting the features from this preprocessed data, and finally using these extracting features to train a machine learning classifier. Through doing so, we were able to create a number of models that were successfully able to classify unseen COVID-19 articles as real or fake. Our traditional machine learning models consist of a Naive Bayes classifier, a Random Forest classifier, a Logistic Regression classifier and a Support Vector Machine. The deep learning models consist of two recurrent neural networks, these were a standard Long short-term memory (LSTM) model, and a bidirectional LSTM. A more in depth analysis on both the traditional machine learning models and deep learning models is provided later in this section, as well as a summary of the results obtained from them.

3.1 Data Preprocessing

For this task, we implemented a standard text preprocessing pipeline in order to transform our original data into a more machine-readable form. We first made use of a regular expression to remove punctuation and make all characters in each headline lower case. After this, each word contained in each article's headline was tokenized, allowing for the removal of stop words, i.e. common words that did not add any additional meaning to the headline. Through doing this, the noise contained in the original data was removed, which allowed our models to classify the articles more efficiently and accurately. Stemming was also applied to the data to reduce the inflectional forms of each word contained in each article's headline, which improved the speed at which our models were able to process the data. We originally considered lemmatization over stemming, however, we found that there was no additional improvement in performance, and as lemmatization is generally a slower process, we ultimately decided against applying it. Figure 1 shows two word clouds which illustrate the most common words used in real headlines (a), and fake headlines (b), after the text preprocessing pipeline had been applied.

3.2 Feature Extraction

For the traditional machine learning models, we decided on a bag-of-words (BOW) approach to extracting features from the text. This meant that each article headline was converted into a bag-of-words vector, which could then be used as an input to our machine learning classifiers. We tested with both term frequency (TF) and term frequency inverse document frequency (TF-IDF) when producing these bag-of-words vectors, which provided a range of different results for our models. Each of these bag-of-words vectors contained unigrams, bigrams and trigrams, as we found that this combination of n-grams gave our models the best accuracy. To add to this, as each headline was relatively short, it was important to extract as much information as we could, whilst ensuring that the models still performed efficiently.

For the deep-learning models, we took a different approach to feature extraction. Rather than using the BOW approach used in the traditional machine learning models, we used both one-hot vector encoding, and the 300-dimensional GloVe pre-trained word embeddings. These feature representations were chosen as we had noticed both have had relative success in similar fields of study, most notably other fake news classification systems such as Ajao et al. [2018]. On top of the standard standard text preprocessing pipeline previously described, we also had to add padding to each headline to allow for a fixed input size to our LSTM models. As the longest headline was 35 words long, each sentence was padded to a length of 35. In the case that a word contained in a headline did not have an equivalent word embedding as a GloVe vector, a 300-dimensional all zero vector was used to represent it.

3.3 Machine Learning Classifiers

By transforming our original data into a BOW vector space, we were able to use several traditional machine learning classifiers to identify real and fake article headlines. Like all supervised learning techniques, each of these models performed an iterative optimization of their objective function during the training process in order to best learn the distribution of real and fake COVID-19 news article headlines. As previously mentioned, the 4 classifiers that were used included: a Naive Bayes classifier, a Random Forest classifier, a Logistic Regression classifier and a Support Vector Machine with a linear kernel. All 4 of these models were trained using TF bag-of-words vectors, and TF-IDF bag-of-words vectors. Table 1 provides a breakdown of the results obtained for each of these models, and displays some of the common evaluation metrics used, such as the accuracy, precision, recall and f1-score.

Table 1: Results obtained from Machine Learning classifiers

Classifer	**Accuracy**	**Precision**	**Recall**	**F1-score**
TF Naive-Bayes	0.850	0.85	0.85	0.85
TF-IDF Naive-Bayes	0.850	0.85	0.85	0.85
TF Random Forest	0.833	0.84	0.83	0.83
TF-IDF Random Forest	0.845	0.85	0.84	0.84
TF Logistic Regression	0.844	0.85	0.84	0.84
TF-IDF Logistic Regression	0.839	0.85	0.84	0.84
TF Support Vector Machine	0.839	0.85	0.84	0.84
TF-IDF Support Vector Machine	**0.856**	**0.86**	**0.86**	**0.86**

As shown above, the best performing model was the TF-IDF support vector machine, which was able to obtain an accuracy of 0.856. On top of this, the TF-IDF support vector machine achieved the highest average precision and recall, and therefore also achieved the highest average f1-score. Interestingly, none of the above models obtained a higher average recall than precision. With this being said, the difference is extremely minor in all cases, and therefore there is no indication of overfitting.

On closer inspection of the classification reports for each model, which are displayed in the iPython notebook, it is shown that all the models, with exception of the naive bayes classifier, achieved a greater precision when classifying real article headlines, and a higher recall when classifying fake article headlines. In relation to the original task, it therefore may be more beneficial to select a model that has a higher precision in classifying fake article headlines, as our ultimate goal was to produce a model which ensured that as much fake news as possible was detected. With this in mind, although the TF naive bayes model achieved a lower overall accuracy, it may be better suited to the task of fake news detection as it obtained the largest precision for classifying fake headlines. Nevertheless, to remain consistent with other related work in literature, we will continue to consider the accuracy of the model as the true indication of its performance.

Following this, we decided to then hypertune the parameters of the TF-IDF support vector machine in a further attempt to increase its accuracy. This was executed by performing an exhaustive search over a number of different parameters of the support vector machine, mainly the regularization parameter and the gamma parameter. Unfortunately, this didn't seem to improve the accuracy of the support vector machine, as it remained at 0.856. Figure 2 displays a normalised confusion matrix for this classifier, allowing visualisation of the evaluation metrics displayed in table 1, and showing how effective it was in classifying the test set article headlines.

3.4 Deep Learning Classifiers

Although a formal explanation of the LSTM is beyond the scope of this paper, we will provide a brief breakdown of what an LSTM is and how they are used for text classification tasks such as identification of fake news.

An LSTM is a type of recurrent neural network (RNN) that is commonly used in tasks such as text classification, generation and summarization. This makes them a good candidate for our proposed task of classifying fake COVID-19 news. Standard RNN's in text-classification suffer from a well-known architectural flaw in deep learning - vanishing gradients. In keeping this description high-level, this essentially means that standard RNN's have a short memory, i.e. they struggle to learn from previously seen information. This is a problem in text classification, as the meaning and context of a word is often conveyed through the use of words prior to it and after it. An LSTM bypasses this problem by storing words in a type of memory that can be accessed at any time. Thus, it is able to learn long-term patterns of words, making it effective in learning context and consequently classifying text. On top of implementing a standard

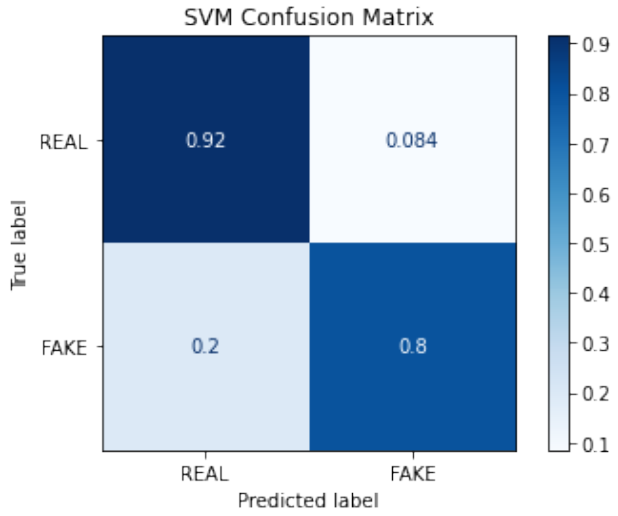

Figure 2: Confusion matrix of the Support Vector Machine

LSTM, we also implemented a bidirectional LSTM, which has a slight architectural difference. Whilst a standard LSTM only takes into account the past context of a word, a bidirectional LSTM uses both the past context and future context by equally considering the words prior to the given word, and the words subsequent, Huang et al. [2015].

As we are performing binary classification between real and fake articles, both the standard LSTM and bidirectional LSTM used a sigmoid activation function and a binary cross entropy loss function. We also experimented with dropout layers, as prior research such as Gal and Ghahramani [2016] has suggested they can improve the performance of LSTM models. In both of our models, the addition of dropout layers improved the overall accuracy, and so we included them in the final architectures. Each model was trained for a total of 50 epochs. Table 2 displays the results obtained from the standard LSTM and bidirectional LSTM, using both the one-hot vector encodings and the pre-trained 300-dimensional GloVe word embeddings. To remain consistent, the same metrics used for the evaluation of the traditional machine learning models are selected, these include the accuracy, precision, recall and f1-score.

Table 2: Results obtained from Deep Learning classifiers

Classifer	**Accuracy**	**Precision**	**Recall**	**F1-score**
One-hot LSTM	0.906	0.91	0.91	0.91
GloVe LSTM	0.844	0.84	0.84	0.84
One-hot Bidirectional LSTM	0.928	0.93	0.93	0.93
GloVe Bidirectional LSTM	**0.950**	**0.95**	**0.95**	**0.95**

As we can see from table 2, the bidirectional LSTM using the 300-dimensional GloVe word embeddings achieved the highest accuracy, at 0.950. It also achieved the highest precision, recall and f1-score, confirming that it was the better model of the two. A normalised confusion matrix for this model is displayed in figure 3.

4 Discussion

Out of all of the models we produced, the bidirectional LSTM obtained the greatest accuracy, achieving a value of 0.950. It is clear that from the results obtained, the deep-learning models seem to be better suited for the task of fake-news classification on this dataset, as both of them provided a fairly significant performance improvement on the traditional machine learning models. Unfortunately, due to the black box nature of neural networks, we were unable to analyse how exactly both the standard and bidirectional LSTM reached conclusions on how to classify the article headlines. On the other hand, by evaluating the decision boundaries produced by the support vector machine, we are able to obtain an insight into the kind of language that it associated with real and fake COVID-19 article headlines. The 10 most "real" and "fake" n-grams that the support vector machine associated with the article headlines is displayed below.

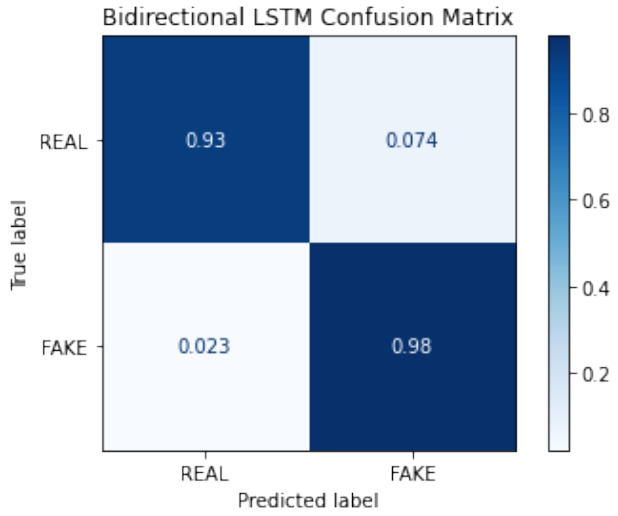

Figure 3: Confusion matrix of the Support Vector Machine

- **Real** - googl, case, say, india, trump, like, coronaviru, app, covid, rate
- **Fake** - claim, govern, brazil, prevent, brazilian, new coronaviru, doctor, solut, video, post

When comparing these to some of the words displayed in the world clouds shown in figure 2, we can notice that the support vector machine was able to correctly pick up on some of the common words associated with the real and fake article headlines. For example, the tokens "googl" and "india" appear very prominently in the real news word cloud, and the token "claim" appears clearly in the fake news word cloud.

5 Conclusion

In this paper, we have demonstrated that a number of different machine learning models, when combined with natural language processing, can be successfully applied to the task of classifying fake news within the COVID-19 domain. To add to this, we have shown that article headlines alone appear to provide a sufficient amount of data to allow for successful classification. This is beneficial as it means machine learning models can be trained in real-time with limited computational overhead. Out of all the models we have produced, the best-performing is the bidirectional LSTM trained on the 300-dimensional pre-trained GloVe word embeddings. This model obtains an overall accuracy of 95%, which we deem high enough to be considered successful. The ever-changing landscape of the COVID-19 pandemic may threat the performance of our models, as they are trained on a fixed knowledge base that may not hold to future findings and statistics. To tackle this issue, the models would need to be systematically trained on labelled data containing newly published real and fake COVID-19 articles. The general lack of labelled data can therefore be seen as a fundamental issue, and is undoubtedly one of the biggest limitations in this work.

A suitable direction for future work would be to increase the size of the current data available by combining the dataset with other existing datasets. As well as this, experimenting with tools such as Google's Custom-Search API could provide to be effective in efficiently web-scraping real and fake COVID-19 articles. On the topic of machine learning, the BERT transformed-based approach to text classification published by Devlin et al. [2019] has seen much success in the field of text classification, and so it would be interesting to measure how it performs in this task, compared to the other state-of-the-art models we have implemented such as the bidirectional LSTM.

We conclude by noting that, as access to major communication platforms such as the internet and social media platforms increases, fake news becomes a more prominent issue. Machine learning can be successfully applied to address this issue, provided there is a rich ecosystem of data available.

References

Sumit Banik. Covid fake news dataset, November 2020. URL `https://doi.org/10.5281/zenodo.4282522`.

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. Fake news identification on twitter with hybrid cnn and rnn models. In *Proceedings of the 9th international conference on social media and society*, pages 226–230, 2018.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. *Advances in neural information processing systems*, 29:1019–1027, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.