Classification of News into Real and Fake News

Group 6

Dhairya Amin
amin.dh@northeastern.edu

Parth Shah
shah.parth2@northeastern.edu

Smit Dar
dar.sm@northeastern.edu

Abstract

Fake news has been surfacing frequently and is pervasive online in recent years as a result of the blooming development of online social networks for various economic and political goals. Online fake news spreads readily through deceptive words, infecting social network members, and has already had a significant impact on offline culture. The prompt detection of fake news is a key objective in enhancing the credibility of information in online social networks. This paper looks into the techniques and programs used to identify false news stories on social media websites and assesses their effectiveness. This research tackles the issues raised by the ambiguous traits of false news and the variety of linkages between news pieces. A variety of techniques, including GloVe embedding, LSTM, RNN, and Logistic Regression, have been developed to train the representations of news items. These techniques are based on a set of explicit and latent features collected from the textual material. The effectiveness of these algorithms has been proven by extensive experiments carried out on a real-world dataset of fake news to compare the performances of various models.

1. Introduction

The rate at which social media platforms such as Facebook, Instagram, and Twitter add new users is alarming. Because social media makes information readily available, it can be a dangerous weapon in the hands of propagandists. Fake news thrives on the spread of scandals, sensationalism, hoaxes, and fabricated stories. The variety and concealment of deceptions make identifying false information in the news difficult, if not impossible. False news can harm people as well as have a negative impact. Changes in the information streams used for news consumption have an impact on people's decision-making processes and how they interpret real-world events. At the organisational level, the impact is greater because it jeopardises brand identities and can alter how their products or services are used. The introduction of news-sharing bots that disseminate unverified material has exacerbated the problem. It is critical to identify false information and fake news on social media and other platforms. There are numerous approaches to dealing with the problem of false information on social media. Statistical methods are used to investigate distribution trends, analyse the source of the information, and understand how various parts of the information are related to one another. Untrustworthy content is classified using machine learning methods, and the accounts that distribute it are investigated. Various techniques concentrate on developing information authentication strategies and specific case studies. The SARS-CoV-2 virus has infected over 190 million people, and WHO has declared COVID-19 a global pandemic. As the system crumbled in many parts of the world during these trying times, people turned to social media and other online platforms to communicate and seek help. People in need of assistance, as well as those providing assistance, found social media platforms such as Twitter to be extremely useful. Because people rely on these social media platforms for news and updates, propagandists began spreading unverified or false information through these channels. As a result, it is critical to developing a quick and accurate model for categorising news. Our team compares models for predicting whether a news article is real or fake to find the best one. Text classification models such as Logistic Regression, RNN, LSTM, and GloVe + LSTM have been used. This project provides a foundation for what model would work best for fake/real news classification, which can then be built upon by news organisations and social media platforms in the future.

2. Literature Survey

Shushkevich and Cardiff, in their paper "TUDublin team at Constraint@ AAAI2021—COVID19 fake news detection (2021)" [2], used an ensemble-based method that combined Bi-LSTM, Support Vector Machine, Naive Bayes, and logistic regression. Their use of logistic regression and Naive Bayes models on the provided fake news dataset produced results that were within 5% of cutting-edge findings.

In their paper on “Covid-19 fake news and hostile post detection in social media (2021)” [3], Sharif, Hossain, and Hoque tested several algorithms, including Bi-LSTM, SVM, CNN, and CNN + Bi-LSTM with embedding techniques including TF-IDF and Word2Vec; SVM with the TF-IDF methodology produced the best results.

Li, Xia, Long, Li, and Li proposed an ensemble model made up of various pre-trained models such as BERT, ERNIE, and RoBERTa, using five-fold five-model cross-validation in their paper “Exploring text-transformers in AAAI 2021 shared task: Covid-19 fake news detection in English (2021)” [4]. Additionally, the performance of the entire model was significantly enhanced by their pseudo-label technique.

An ensemble approach for detecting fake news that is driven by heuristics was proposed by Das, Basak, and Dutta in their paper on “A Heuristic-driven Uncertainty based Ensemble Framework for Fake News Detection in Tweets and News Articles (2021)” [5]. They used an ensemble model made up of pre-trained models as well as a statistical feature fusion network, a new heuristic technique, and several characteristics found in tweets or news items that could be used as statistical features, such as URL domains, username handles, and source. The efficacy of the methodology was demonstrated using the fake news datasets from COVID-19 and FakeNewsNet. On the FakeNewsNet dataset and the COVID-19 dataset, they got the best results, with an F1 score of 0.9073 and 0.989, respectively.

3. Dataset

The “Fake and real news dataset” [1] is derived from Kaggle. The dataset consists of 4 columns (features), the title of the article, text of the article, subject of the article, and the date when the article was posted. The dataset has been confirmed and is drawn from a variety of websites and social media channels. The dataset classified as "real" was compiled from reputable sources that provided accurate and realistic information. The data that has been dubbed "fake" was gathered from articles, tweets, and postings that contained false information. This dataset includes roughly 40,000 articles that include both real and fraudulent news. The information about fake and real news is provided in two distinct datasets, each of which contains about 20,000 articles. In addition to that, the fake news dataset consists of 39% news, 29% are political and the rest 32% are labelled as others. These articles have been posted from 30 Mar 2015 to 18 Feb 2018. In the real news dataset, there are 53% political articles and the rest 47% are world news. These articles have been posted between 12 Jan 2016 and 30 Dec 2017. There is no need to improve this dataset to make it more balanced because it is reasonable to presume that it is balanced already. The count plot of the data labels is shown in Figure 1. It depicts the data counts in each category graphically. The vocabulary size of the dataset is around 1,22,248, wherein around 90% of the articles have less than 700 words, which is graphically shown in Figure 2.

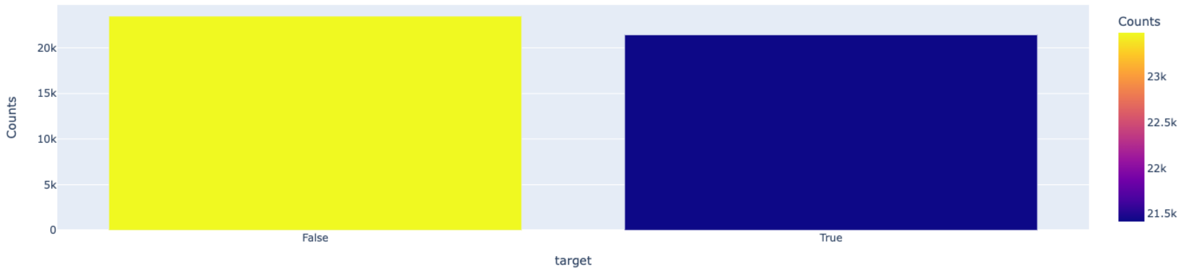


Figure 1

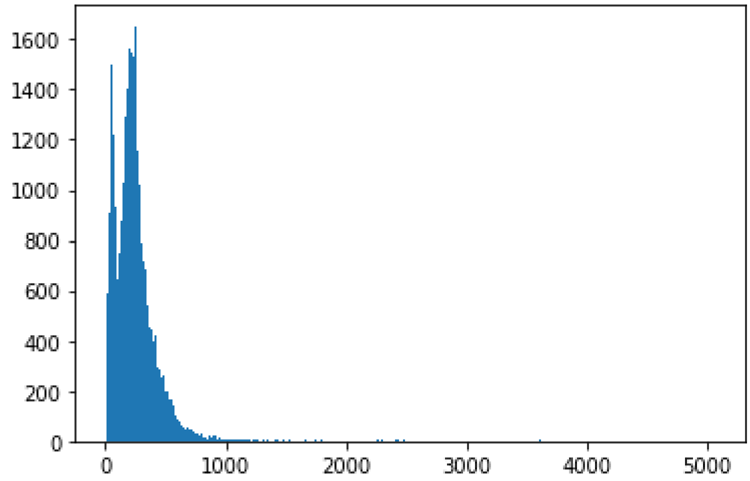


Figure 2

4. Method

We propose a three-stage process that starts with preprocessing, in which raw data from the website is cleaned and prepared for feature extraction, the cleaned text is converted to vectors or word embeddings, and pre-trained models like GloVe, Word2Vec, and others are used. To gain a better understanding of the dataset, we perform exploratory data analysis using techniques such as data imbalance, word cloud, statistical analysis using various plots, different n-grams analysis, feature extraction using count vectorizer, TF-IDF vectorizer, and Hashing Vectorizer, among others. The approach determined the dimensions of each word. Several deep learning models were used to process the vectors, including the traditional Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Logistic Regression. With one neuron output layer, sigmoid nonlinearity was used in the output layer. The Adam optimizer was used with different learning rates based on the model. We evaluate these models by performing metric comparisons on them to determine the best technique for achieving the highest accuracy on the available dataset. We use confusion metrics like accuracy, precision, recall, the area under the curve, and so on. We want to achieve the best possible results, preferably above 95%. The entire project is written in Python and runs on Google Colaboratory. TensorFlow, Pandas, NumPy, Matplotlib, Seaborne, Natural Language Toolkit (NLTK), and other libraries are used. Each stage is described in detail below.

4.1 Pre-processing

We combined the true news dataset with the fake news dataset and set the target variable to 1 for true news and 0 for fake news. Data preprocessing is an important step in any NLP task. To obtain cleaned data, the raw data collected through various means is passed through a pipeline of NLP preprocessing. This pipeline includes the following NLP tasks:

- Removing newlines and tabs/whitespaces/links/special characters/ stopwords
- Strip HTML tags
- Reducing repeated characters and punctuations
- Expand contractions
- Correcting misspelt words
- Lemmatization/stemming

4.2 Exploratory Data Analysis

Exploratory Data Analysis was performed on the dataset to gain a better understanding. These include word clouds for the real and fake datasets, as shown in Figures 3 and 4, respectively. Figure 5 graphically depicts the categories of news articles. Figures 6 and 7 show the results of some statistical analyses of the dataset, such as the number of words in each article and the average word length. Finally, we examined the dataset using n-grams such as unigram, bigram, and trigram. Figures 8 through 10 show these graphically.

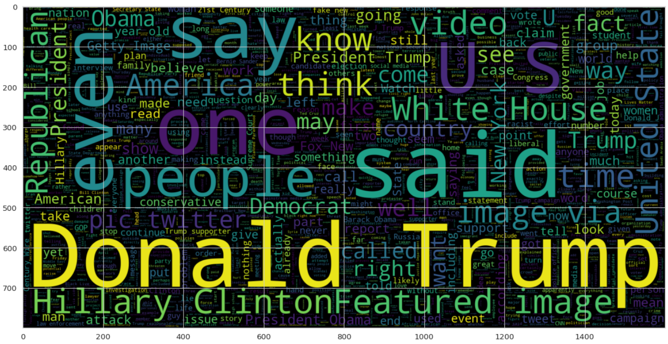


Figure 3

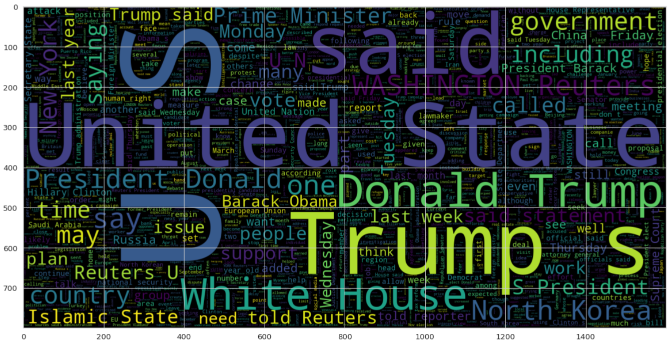


Figure 4

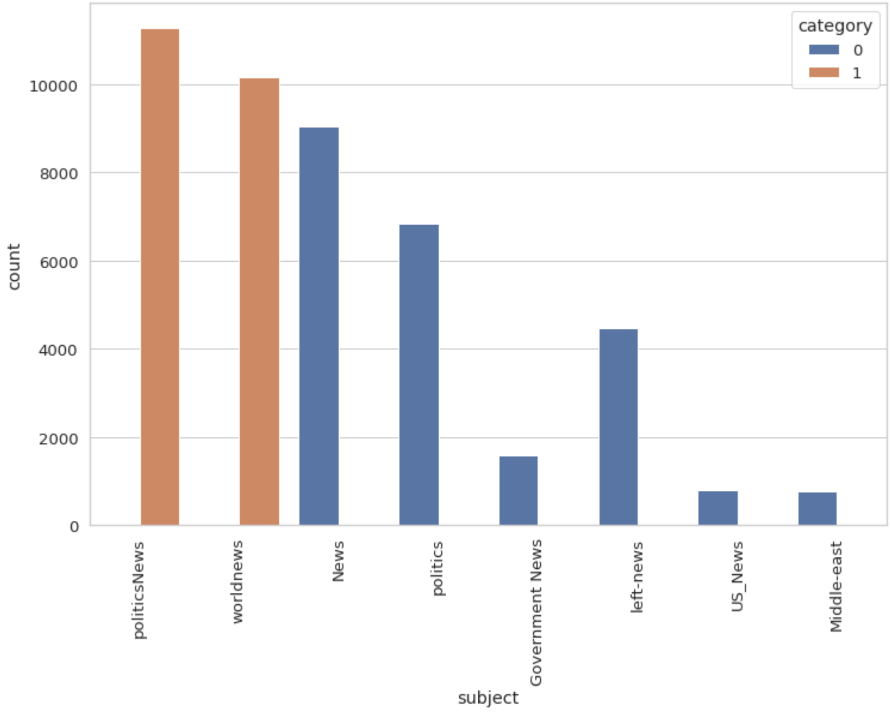


Figure 5

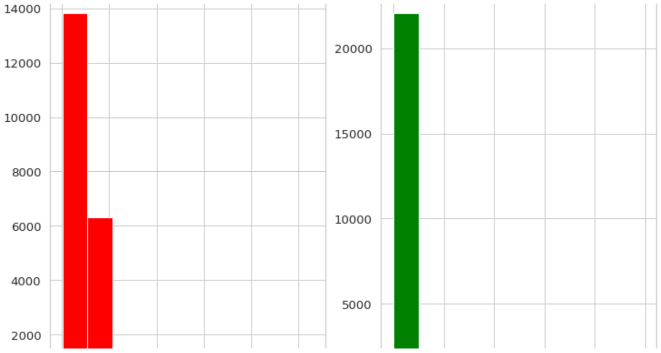


Figure 6

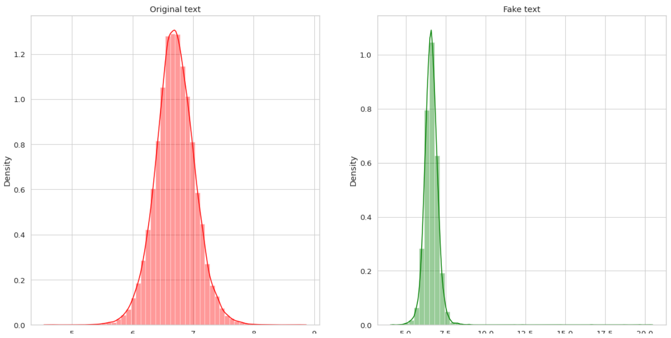


Figure 7

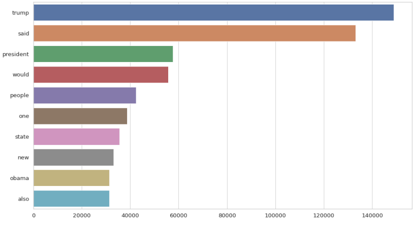


Figure 8

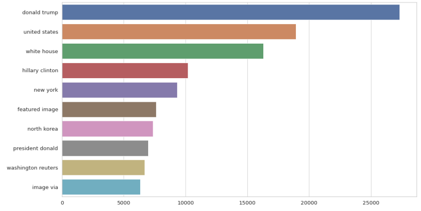


Figure 9

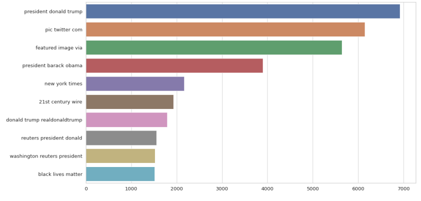


Figure 10

4.3 Word Embeddings

Word embeddings are vectorized representations of text that have similar representations for words with similar meanings. Before feeding the text to a machine learning model, all tokens in the text must be converted to word embeddings. While training the model, the vector's dimension can be passed in as a hyperparameter. There are numerous methods for converting text to vector; commonly used models include BOW, TF-IDF, Word2Vec, and GloVe. We have primarily focused on GloVe and Word2Vec embeddings.

GloVe - GloVe, or global vector, is a word embedding model developed by Stanford University researchers in 2014. GloVe generates vector representations by using the similarity score between two tokens as an invariant. The model employs two distinct model techniques: the skip-gram model and the continuous bag of words model (CBOW). The GloVe model is based on the old theory of word or token co occurrence and was presented as an improvement to Mikolvo's existing model Word2Vec. GloVe generates word vectors using both global and local statistical data. The GloVe model is capable of accurately capturing semantic information. The main disadvantage of the skip-gram model was its high computational cost and time (despite its high accuracy), whereas CBOW had low accuracy (although low computational time). GloVe attempts to combine the best of both worlds and has produced accurate results in a short amount of time. The proposed work made use of the GloVe pre-trained model. For better results, a 300-dimension vector was chosen. GloVe was able to vectorize the majority of the tokens in the text. However, in the training process, a few tokens that were not converted to vector form were omitted.

Word2Vec - Word2Vec's effectiveness stems from its ability to group vectors of similar words. Word2Vec can make good guesses about a word's meaning based on its occurrences in the text given a large enough dataset. Word associations with other words in the corpus are produced by these estimates. Words like "King" and "Queen," for example, would be very similar. You can find a close approximation of word similarities by performing algebraic operations on word embeddings. For example, the 2-dimensional embedding vector of "king" - the 2-dimensional embedding vector of "man" + the 2-dimensional embedding vector of "woman" yields a vector very close to the 2-dimensional embedding vector of "queen." The success of Word2Vec is due to two main architectures, CBOW, and skip-gram architectures.

4.4 CountVectorizer + Logistic Regression

Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analysing the relationship between one or more existing independent variables. In this approach, the dataset was randomly divided into 80:20 train and test sets. The Count Vectorizer was used to convert text data into a token count matrix. Three different predictions are made, each with a different component, and their outcomes are analysed using the accuracy and confusion matrices. We made predictions based solely on the title. Following that, predictions are made using only the contents of the articles. Finally, the prediction was based on the combination of the article titles and content.

4.5 RNN

Recurrent Neural Networks are a type of artificial neural network that is commonly used for sequence data such as natural language or time series data. RNNs are Neural Networks in which the previous layer's output is also supplied as input to the next layer, assisting the network in better understanding the relationships. At each element of the sequence, the model examines not only the current input but also what it knows about the previous elements. This memory enables the network to learn long-term sequence dependencies. In this approach, the words are sequenced to create the word vectors. The data is then randomly split into 80:20 train:test ratio. The model consists of an embedding layer, two bidirectional LSTM layers, one dropout layer, and two dense layers, utilising the ReLu activation function. This model yielded impressive results.

4.6 Word2Vec + LSTM

Long short-term memory (LSTM) networks are commonly used to replace RNN networks to avoid problems such as vanishing and exploding gradient descent. LSTMs have also been shown to be effective at remembering long-term dependencies that RNNs could not. The LSTM cell is composed of three gates: the forget gate, the input gate, and the output gate. LSTM also maintains a cell state that transports information across the cell with minimal linear changes. We compared the prediction results of manually created word vectors versus pre-trained vectors trained on a subset of the Google News dataset for this approach (about 100 billion words). Three million words and phrases are represented by 300-dimensional vectors in the model. The LSTM model for the first part, where the word vectors are created using gensim, consists of an embedding layer, an LSTM layer, and the final dense layer. Finally, the model that used the pre-trained vectors had one layer of embedding, convolutional, max-pooling, LSTM, and final density each.

4.7 GloVe + LSTM

In this approach, the GloVe embedding is employed. The data is shredded off of the title, subjects and dates. The learning rate for this model is intelligently reduced when the model stops learning anything new. Here, the dataset is randomly split and passed through an LSTM model that consists of an embedding layer, two each of LSTM and dense layers, and uses the ReLu and Sigmoid activation functions. This model, with the inclusion of the GloVe embedding performs exceptionally well.

5. Results

The proposed methods were iteratively trained on the dataset to reduce the loss function and increase the accuracy. The accuracies of various models are shown in Table 1.

Model	Accuracy
CountVectorizer + Logistic Regression with only titles	94.75%
CountVectorizer + Logistic Regression with only contents	99.54%
CountVectorizer + Logistic Regression with both titles and contents	99.67%
RNN	98.89%
Word2Vec + LSTM with manual vectors	97.49%
Word2Vec + LSTM with pre-trained vectors	97.98%
GloVe + LSTM	99.74%

Table 1

The confusion matrices and values of these models have been summarised as follows:

- **CountVectorizer + Logistic Regression**

  The three confusion matrices for each component/condition are depicted in the order from left to right.

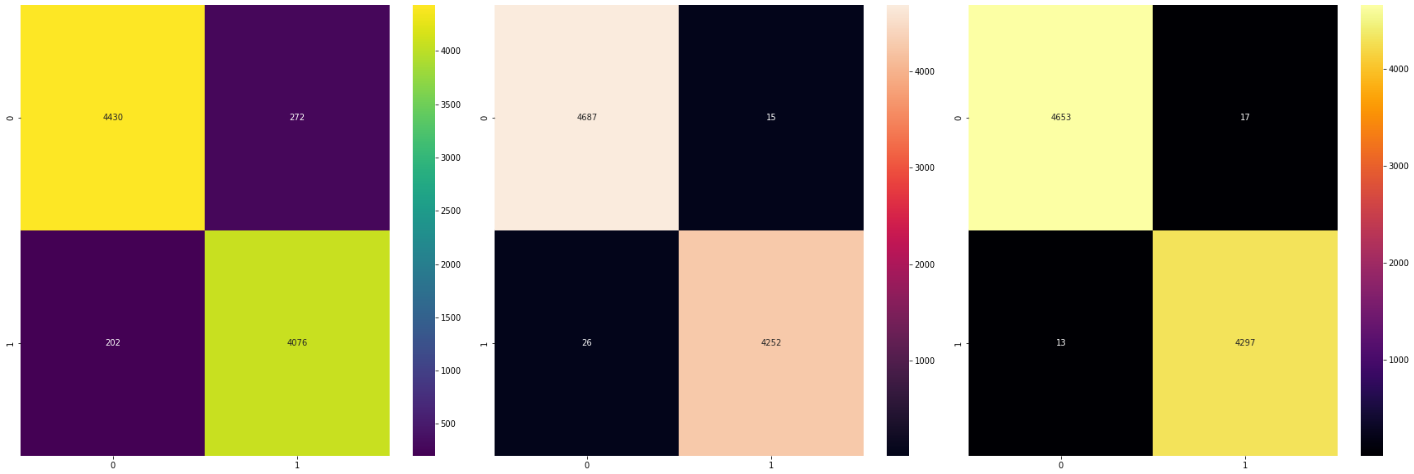


Figure 11

- For the predictions based solely on titles (Figure 11 - left), 4430 fake news articles were classified as fake, while 4076 real news articles were classified as real. However, we still have 474 titles that are misleading and incorrectly classified.
- For the predictions based solely on contents of the articles (Figure 11 - middle), 4687 fake news articles were classified as fake, while 4252 real news articles were classified as real. However, we still have 41 article contents that are misleading and incorrectly classified.
- For the predictions based on the titles and contents of the articles (Figure 11 - right), 4653 fake news articles were classified as fake, while 4297 real news articles were classified as real. However, we still have 30 articles that are incorrectly classified.

Hence, here we can conclude that when considering the titles and the contents together gives the highest accuracy on the model.

- **RNN**

This model gave impressive results:

- Accuracy - 98.89%
- Precision - 99.05%
- Recall - 98.64%

The visualisation of training over time is shown in Figures 12 and 13.

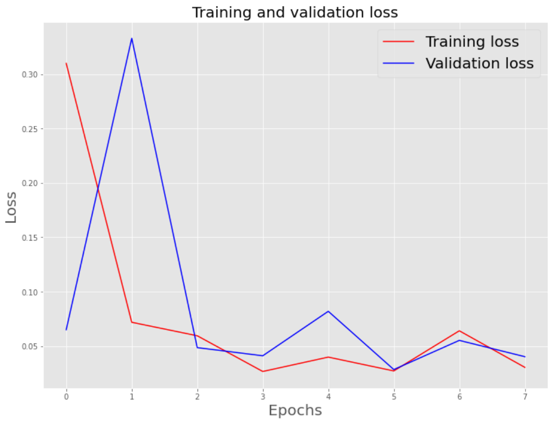


Figure 12

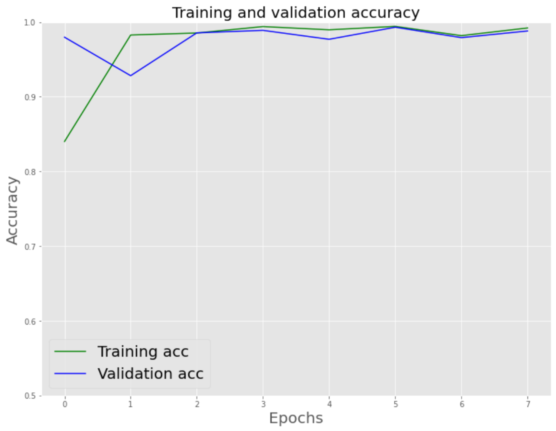
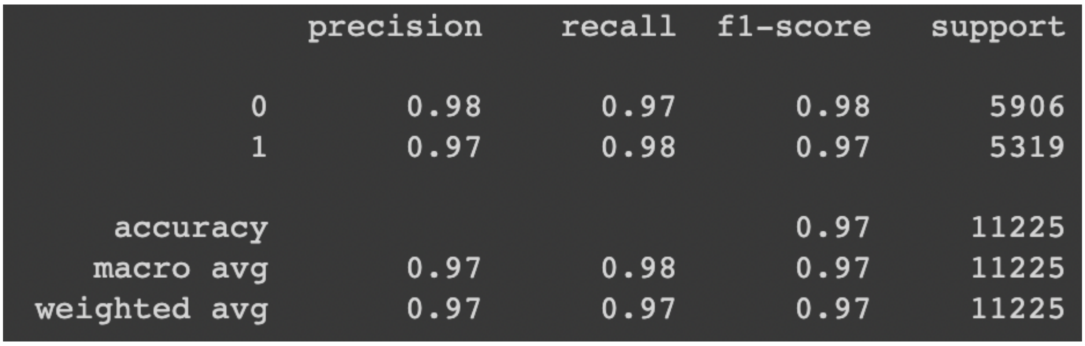


Figure 13

- **Word2Vec + LSTM**

This model gave an accuracy of 97.49% when manually created word vectors were used. It produces the following statistical performance:



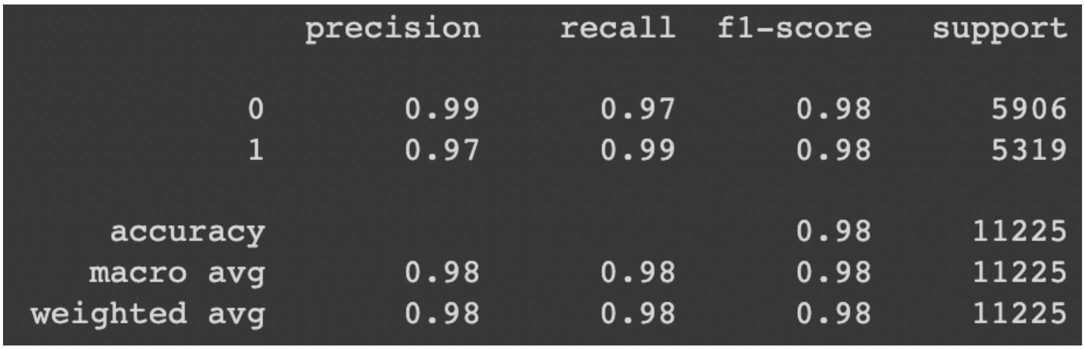
```
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      5906
           1       0.97      0.98      0.97      5319

    accuracy                           0.97     11225
   macro avg       0.97      0.98      0.97     11225
weighted avg       0.97      0.97      0.97     11225
```

Figure 14

This model gave an accuracy of 97.98% when pre-trained word vectors were used. It produces the following statistical performance:



```
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      5906
           1       0.97      0.99      0.98      5319

    accuracy                           0.98     11225
   macro avg       0.98      0.98      0.98     11225
weighted avg       0.98      0.98      0.98     11225
```

Figure 15

These results suggest that the pre-trained word vectors perform marginally well.

- **GloVe + LSTM**

This model outperforms all the other models and approaches, with an exceptional accuracy of 99.74%. The accuracy and loss of the models over the epochs is depicted in Figure 16.

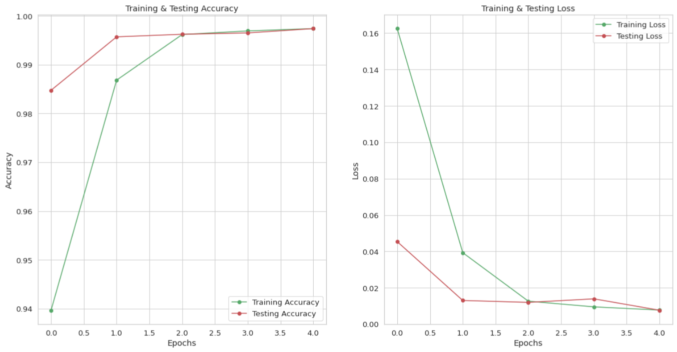


Figure 16

The confusion matrix for the same is given in the following image.

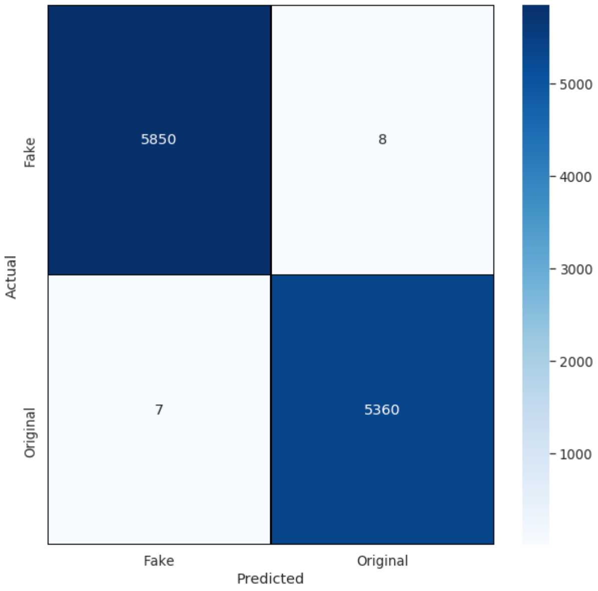


Figure 17

Here, 5850 fake news articles were classified as fake, while 5360 real news articles were classified as real. However, we still have 15 titles that are incorrectly classified.

6. Conclusion

This work aimed to classify fake news from real news related to various categories. A fake and real news dataset from Kaggle was used to train and test the models. Various supervised machine learning models such as Logistic Regression, RNN, and LSTM were trained, in inclusion with word-embedding techniques such as CountVectorizer, GloVe, and Word2Vec, to compare the performances. The GloVe + LSTM model out-performed other models. Although the GloVe + LSTM model outcasted other models, it took a much longer time to train than the rest of the models, but a huge difference in accuracy was not seen. The logistic regression model trained the quickest, taking almost half the time as the RNN and Word2Vec + LSTM networks, and one fourth the time of a GloVe + LSTM network.

7. Future Work

Future work could aim to improve the accuracy of the models even further. Convolutional Neural Networks can be used in conjunction with other sequential models. Transformers, BERTs, and GPTs are attention networks that can be fine-tuned to classify fake news. We can also consider domain-specific approaches to data modelling, such as NewsBERT. It can also be beneficial to use extended datasets, which may include more than one dataset.

8. Acknowledgment

This project was completed for CS6120 under the instruction of Professor Uzair Ahmad.

References

[1] Dataset - https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset
[2] Python Notebooks - https://github.com/parth2608/Fake-News-Detection

[3] Shushkevich E, Cardiff J (2021) TUDublin team at Constraint@ AAAI2021—COVID19 fake news detection. arXiv preprint https://arxiv.org/abs/2101.05701

[4] Sharif O, Hossain E, Hoque MM (2021) Combating hostility: Covid-19 fake news and hostile post detection in social media. arXiv preprint https://arxiv.org/abs/2101.03291

[5] Li X, Xia Y, Long X, Li Z, Li S (2021) Exploring text-transformers in aaai 2021 shared task: Covid-19 fake news detection in English. arXiv preprint https://arxiv.org/abs/2101.02359

[6] Das SD, Basak A, Dutta S (2021) A heuristic-driven uncertainty based ensemble framework for fake news detection in tweets and news articles. arXiv preprint https://arxiv.org/abs/2104.01791

[7] Kulkarni, C., Monika, P., Shruthi, S., Deepak Bharadwaj, M.S., Uday, D. (2022). COVID-19 Fake News Detection Using GloVe and Bi-LSTM. In: Shakya, S., Du, KL., Haoxiang, W. (eds) Proceedings of Second International Conference on Sustainable Expert Systems . Lecture Notes in Networks and Systems, vol 351. Springer, Singapore. https://doi.org/10.1007/978-981-16-7657-4_5