

ISSS610

Applied Machine Learning

Final Report:

Fake News Filter

Group 18

He Xinyi, Evelyn

Tang Mingzheng Paul

Madeline Lim

Lim Chuan Ren Jonathan

Sam Yong Zhi

## Introduction

As the world goes more connected and digital, the spread of fake news becomes an ever-growing threat to society and corporations. The business impact of fake news is especially huge to social media platforms, news agencies and large corporates, as it causes not only reputational and PR crisis, but also monetary damages. This project aims to solve the business problem by building an effective Fake News Filter to automate the detection of fake news. Various approaches, such as traditional machine learning, information retrieval and cross reference, as well as deep learning, have been experimented on. The different models yield varying results, which will be discussed further.

## Dataset

Several datasets were considered for this project, including a Kaggle dataset[[1]](#footnote-2), and the Liar dataset[[2]](#footnote-3) compiled by Wang et al. (2017). However, each dataset suffered from its own set of drawbacks. For example, the identified Kaggle dataset had several formatting issues, and did not have clear information on the source of the data. Moreover, both of the above datasets were also limited in scope to only US politics.

Our final chosen dataset is a combined corpus[[3]](#footnote-4) prepared by Khan et al. (2021) containing the full text of 80k news articles, as well as a real (1) or fake (0) label for each article. Although other information (e.g., source, article title) was not available, a key differentiating factor of this dataset is that it incorporates a wide range of topics, including politics, economy, healthcare, and sports. In addition, the use of the dataset in a paper published by a credible journal also lends some credibility.

In the paper, the authors described collecting the articles mostly from 2015 to 2017, and provided examples of real and fake news sources. Real news sources include The New York Times, CNN, Reuters and Business Insider, while multiple types of fake news such as hoax, satire and propaganda were collected from sources such as The Onion, Borowitz Report, American News and Natural News.

## Pre-processing

In this project, data pre-processing was minimal. We used the provided train and test sets of the combined corpus as split by the authors. For some of the models which are affected by the length of sequence (LSTM), we removed the stopwords to limit the length and allow it to cover more context over fewer neighbouring tokens.

## Models

In this project, we experimented with both traditional and deep learning models to classify a given text input as real or fake. In addition, we also explored implementing a cross-reference model, which aims to identify top-k most similar texts based on a given text input.

Although pre-trained transformer models such as BERT have been shown to produce state-of-the-art results on numerous NLP tasks in recent years, we have chosen not to solely focus on arriving at the highest accuracy scores. As such, we have also explored various traditional machine learning methods, which allows for greater explainability of the model results.

### Traditional Machine Learning Models

#### Logistic Regression with Feature Extraction

For our most basic model, we first extracted various features such as word count, average word length and average sentence length, and performed some exploratory data analysis to visualise the differences in characteristics between real and fake news articles. A logistic regression model was then built based on these extracted features.

#### Logistic Regression, Support Vector Classifier and Naive-Bayes with Count Vectorizer / TF-IDF

We implemented using scikit-learn and trained three classifiers – Logistic Regression, Support Vector Classifier, and Naive-Bayes to predict the fake news. Our plan is to use these models to establish some form of baselines as comparisons for the deep learning models, thus we used the default parameters with minimal feature engineering and hyper-parameter tuning. First, we obtained the term frequencies and count vectorizer that will be included as input attributes for the classification models. To bind the count vectorizer, TF-IDF and classification model together, we made use of the pipeline function to help automate the workflows.

### Cross Referencing Model

#### MiniLM v6 Model

Using the pretrained MiniLM v6 model, we generated text embeddings for each article which contains the contextual meaning. We treat the training set as a database of labelled real and fake articles and generate a text embedding vector for each article in the training set. To predict whether an article in the test set is real or fake, we first generate the text embedding vector for this candidate article. We then obtain the top k most similar articles in the training set using cosine similarity and use majority voting among the retrieved articles to determine whether the candidate article is real or fake.

### Deep Learning Models

#### Bidirectional LSTM with GloVe Word Vectors

As fake news classification can be formulated as a textual sequence classification task, we are interested to explore the performance of recurrent neural network (RNN) on it. Word embeddings were initialised with 200-dimensional pre-trained embeddings from GloVe. We then implemented a bidirectional long-short term memory model (LSTM), via tensorflow keras. LSTM is an RNN unit able to control informational flow along time. A bidirectional approach was chosen to better capture textual context from both beginning and future. The model was implemented with 128 units and a dropout of 0.3. A dense layer served as output layer with sigmoid activation function for binary output. It was trained with Adam as the optimizer for 3 epochs, validated on the validation set.

#### XLNet Model

We used the state-of-the-art XLNet model to obtain our best results. This model utilises attention mechanisms and transformers, which allows the model to retain better long distant relationships between tokens in the sequence. XLNet also uses BPE for to model subwords, which allows the model to draw meanings between similar words even if they are new to the corpus. With the addition of adding recurrent segments between blocks of 512 tokens, which are usually processed separately in BERT, this allows the model to have a better understanding of very long text sequences. The model was trained over 5 epochs, with each epoch taking 110 min, using the AdamW optimizer.

## Evaluation Metrics

For this classification task, we primarily used accuracy scores, precision, recall and F1-score. We used the weighted F1-score due to the imbalanced data of real versus fake news in the provided test set. We also compared the training time for each model to better understand the trade-offs between training efficiency and performance.

## Results

The model performance results are summarised in the table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Classifier** | **Training time** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| LR + Features | <5s | 73% | 0.72 | 0.73 | 0.72 |
| SVC + Count / TF-IDF | 60s | **91%** | **0.91** | 0.91 | 0.91 |
| LR + Count / TF-IDF | 90s | 90% | 0.90 | 0.90 | 0.90 |
| NB + Count / TF-IDF | 20s | 88% | 0.88 | 0.88 | 0.88 |
| Cross Ref | 150s | 75% | 0.77 | 0.75 | 0.76 |
| Bi-LSTM + GloVe | 1200s | 68% | 0.66 | 0.68 | 0.67 |
| XLNet | 550min | 90.8% | 0.89 | **0.99** | **0.94** |

## Analysis

### Logistic Regression with Feature Extraction

The EDA performed on the extracted features did not yield much noteworthy observations, as there were no significant observable differences between real and fake news on most individual features. The exception was the number of words in real vs fake articles – compared to fake news articles, real news articles had a much higher average word length and longer tail of more wordy articles.

Given these observations, the logistic regression model performed surprisingly above expectations based on these extracted features, with an accuracy score of 73% on the test set. The standardised weights of the various features are shown in the figure below.

Chart, bar chart

Description automatically generated

An analysis of the weights indicates that fake news articles are correlated with having a shorter number of words, longer average word lengths, and greater proportion of exclamation and question marks. These findings largely corroborate with our perceived impression of fake news articles, which may at times be more inflammatory and opiniated (hence the use of more of such punctuation marks).

While this model is definitely limited in its predictive power, it nevertheless is able to provide us with some insights as to how the characteristics of real and fake news articles differ at a high level.

### Logistic Regression, Support Vector Classifier and Naive-Bayes with Count Vectorizer / TF-IDF

All three classifiers performed and achieved similar accuracy of 88% to 90% on the test set. The performance is especially impressive considering that there were minimal feature engineering and hyper-parameter fine tuning. Moreover, these models trained extremely fast, longest being Logistic Regression taking 90 seconds and Naïve Bayes Classifier took a mere 20 seconds.

To get some interpretability of these models, we used ELI5 get importance of features globally and locally. ELI5 is a Python library which allows to visualize and debug various machine learning models using unified API. It supports all the scikit-learn algorithms (i.e., algorithm that supports .fit & .predict methods).

The tables below give us the weight associated to each feature for our Logistic Regression model and SVC model. The value tells us how much of an impact a feature has on the predictions on average, the sign tells us in which direction.

|  |  |
| --- | --- |
| **Logistic Regression** | **SVC** |
| Timeline  Description automatically generated |  |

We noticed the words “according”, “reportedly”, “sources”, “reports” are associated with fake news with heavy weights, which make sense as fake news likely had to use such words to gain credibility. On the flipside, the word “but” turned out to be the top weighted word for true news, perhaps true news is more likely to present more balanced views, presenting contrast and opposing views within the article compared to fake news.

In terms of local interpretation, as visualised below, we can see how a particular news article (which is true) has been classified as fake by the classifier. Essentially our models work by summing up all the weights to determine the final prediction. True news with writing style that somehow contained too many negative words with a net score that makes it classified as fake news does happen, which unfortunately is the limitation of the models.

Text

Description automatically generated

### Cross-Referencing Model

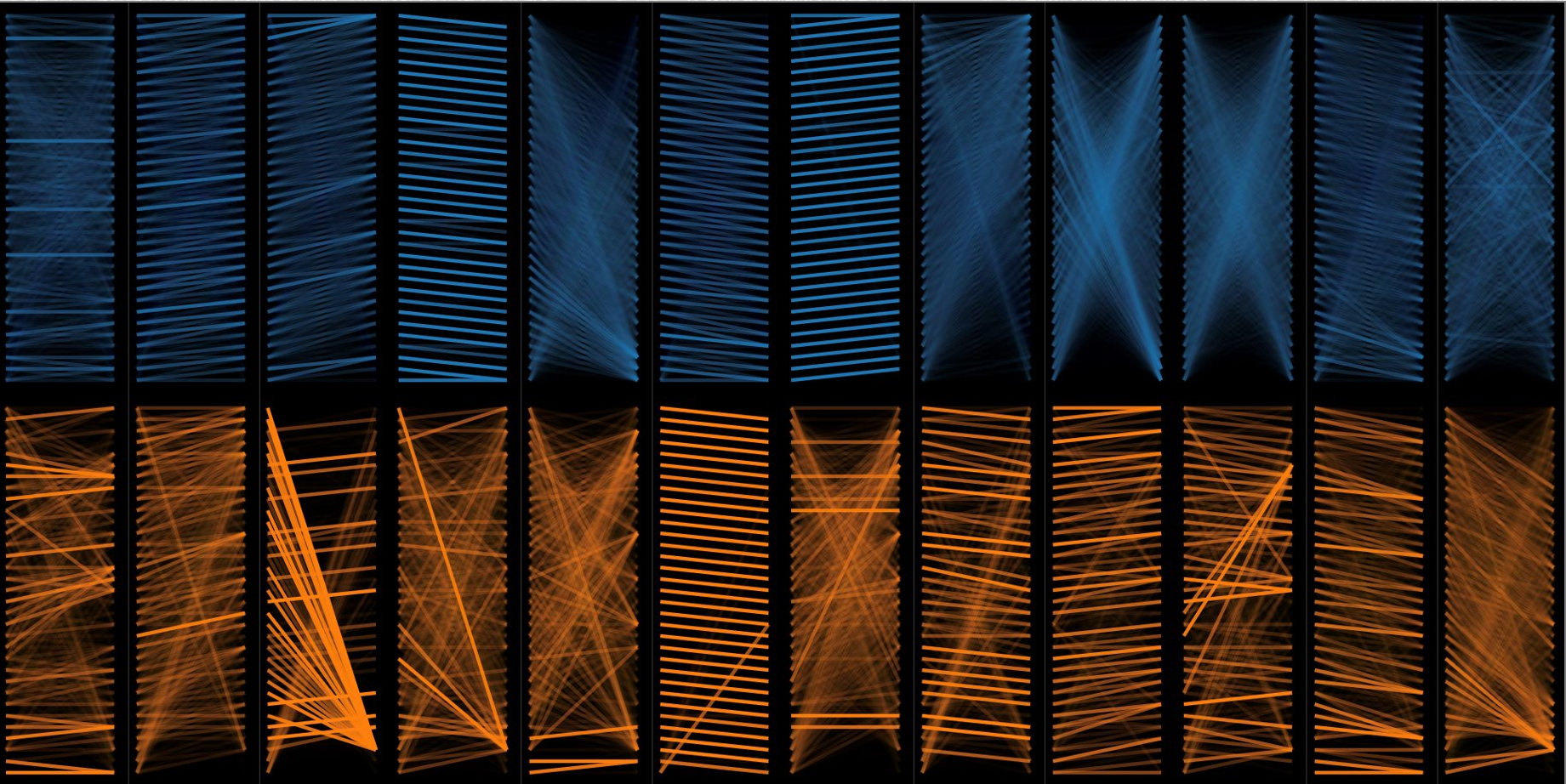
The cross-referencing model performed poorly on the test set relative to the traditional machine learning models tested. However, within the training set the pretrain MiniLM model obtained an accuracy of 86%. This could be due to the fact that the cross-referencing model does not take into account the stance taken within the text. The text embedding produced by MiniLM may only include the general topics mentioned within the text but does not learn whether it agrees / disagrees with the topic discussed. This model depends heavily on whether the content in the candidate article is well covered in the training set used as a database. This could be another reason why this model performs less well compared to the traditional machine learning models mentioned earlier.

### Bidirectional LSTM

The model performance was only able to achieve a mediocre weighted F1 score of 0.67. A possible factor is that the length of input tokens, after pre-processing, is still too long for the model to capture the full context. In the current dataset, there could be as much as 4000 tokens in one news article. In addition, to limit the number of words in the input sequence we removed stopwords, which were identified to have significant differences between the fake and real articles (e.g., “but” and “it”).

### XLNet

XLNet achieved the best performance with an F1 score of 0.94. With its ability to handle long sequences, the removal of stopwords was not needed and it could fully utilise the input sequences while minimizing information loss. To better understand how the attention mechanisms operated, we used the BertViz library to visualise the attention weights given an example input sequence.



The attention weights in the figure above show 2 of the attention heads across multiple layers. Looking at the weights we can see that while some layers employ a simple direct connection from tokens immediately before / after, we can also see how it models relationships between tokens very far away from each other.

Conclusion

The use of a variety of traditional and deep learning models, as well as a cross-reference model, has allowed us to achieve high performance, while also offering explainability behind the results obtained, which can often be a challenge when using deep learning models.

While we achieved the best F1-score from the XLNet model, we were able to glean significant insights from the more explainable traditional machine learning models. From our basic logistic regression model with extracted features, we were able to determine the effect of overall characteristics of text articles on their eventual classification. In addition, our models using count vectorizer and TF-IDF could help us determine keywords that differentiate between real and fake articles. This showed some interesting results where fake articles attempt to look credible using words such as “sources” and “according”. Real articles on the other hand displayed a more objective and balanced stance using words like “but” and “related”.

Finally, the cross-referencing model can utilise existing known real / fake articles and compare them with a candidate article to determine whether the candidate article is real or fake. While our cross-referencing model did not perform as well as the traditional machine learning models, it can be argued that the model lacks sufficient data and performs poorly when the candidate article has content that is not well-covered by the training set. In addition, with a different dataset and model we could train it to learn not only the topics mentioned within the text but also the sentiment and stance taken regarding the topic.

## References

Khan, J. Y., Khondaker, M. T., Afroz, S., Uddin, G., & Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. Machine Learning with Applications, 4, 100032. <https://doi.org/10.1016/j.mlwa.2021.100032>

1. <https://www.kaggle.com/c/fake-news/data> [↑](#footnote-ref-2)
2. <https://www.cs.ucsb.edu/~william/data/liar_dataset.zip> [↑](#footnote-ref-3)
3. <https://github.com/JunaedYounusKhan51/FakeNewsDetection> [↑](#footnote-ref-4)