Classification and Prediction of Fake News Through Trained, Generated Phrases

Rommel Lantajo II and **Riik Acharya**
Vassar College

Abstract

Most people online only read the news based on the title itself, but that title can be misleading or article itself may be full of inaccuracy. This paper's focus is if we can reduce the title to only it's key phrase with YAKE and examine it within DistilBert's classification to see if a key phrase is an accurate indicator of a fake new article. This process takes a dataset, extract it's key phrase and YAKE score value, and gives a predicted classification from DistilBert, with that YAKE score, f1 score, and accuracy with it's actual and predicted label.

1 Introduction

Fake News is a prevalent problem plaguing our society today. Especially with the rise of social media, it's easier than ever to spread fake news to a wide variety of unsuspecting viewers. Not only that, but because of social media, it is easier for people to retreat into their own bubbles and only consume news that they know already backs up their own viewpoints. There had also been a rise of people not reading the actual article but instead only the title itself and make their own conclusions from it. But are there key phrase within the article that is can be used as an indication that the article is authentic or fake? This paper proposes a methodology to create a model that can detect if an article's title classification through a simplification of the text to only it's key phrase. Then, that key phrase, that was manually classified, is passed through that model and test it's classification against the manually classification alongside the probability of it being actual key phrase to be plotted accordingly.

2 Related Works

There have been many previous studies done to try and build a fake news detection system over the years.

One study (Choudhary and Arora, 2021) has explored many methods of identifying features and semantics of fake news. One of them was tracking word density, as fake news articles tend to use certain words more often. They also explored sentiment based features of each statement like Polarity and Subjectivity, classifying arguments based on positive and negative words as well as how opinionated they were. They extracted the grammatical features of each sentence such as noun, verb, adjective, pronoun, and adverb to see if any of them contained any certain words. They also assessed the readability of each article, making sure the person reading it could fully understand it. We are also examining features of real and fake news titles, but only by the perceived key phrases.

There were also a group of researchers (Aldwairi and Alwahedi, 2018) that built an extension that, when the user does a web search, identifies sites with links that may mislead them. They flagged sites that had a lot of hyperbole and slang phrases because those as well as articles with an excessive amount of exclamation points because those they hypothesized, were more likely to be fake. The most important feature is that it flagged articles whose titles were too distinct from their actual content because these were "clickbait".

Zhanam and others also did a study (Z Khanam and Rashid, 2021), they are also researching the exact same problem to different algorithms and their performance. Specifically, they are attempting to turn the data through multiple different classification algorithms like the Random Forest decision tree algorithm, Support Vector Machine, Naive Bayes, and KNN in order to build their model. However, the main focus on our is not the classification performance, but a reduction of text in comparison to their accuracy in a single model. Which is why we built our model through DistilBert.

Other researchers (Waikhom and Goswami, 2019) are also researching the exact same problem but with the LIAR dataset as the training data.

They are specifically using n-grams converted into TFIDF vectors, labeled, then processed through flatting and normalization. The features are also labeled numerically and categorically to help with scaling and accuracy.

Many studies have found that combining multiple methods have yielded better results, such as this one study (Iftikhar Ahmad, 2020). This study combined a bunch of different methods together to produce four combined models, and those models ended up dominating the benchmark algorithms such as just different types of Neural Network, because combining these different methods addressed the problems each of these algorithms had on their own. This is relevant to us because we are planning on combining multiple different methods. However, we are planning on using transformers rather than using methods such as Random Forest and Voting Ensemble Classifiers. Aldwairi's group (Aldwairi and Alwahedi, 2018) also used four types of classifiers for fake news prediction: Bayesian Net, Logistic, Random Tree, and Naive Bayesian Net. For all four of these classifiers, they achieved almost flawless results in precision, recall, F-score, and ROC. Naive Bayes itself didn't perform the absolute best but didn't perform the worst. Our study only focuses on the precision, recall, F-score, and ROC to visualize our data and test the accuracy of the generated phrases from title created through YAKE.

Various studies have actually found that transformers work better than most other machine and deep learning models. One such study, actually combined 3 transformer models: BERT, ALBERT, and XLNext, by taking the softmax of all of them, and then averaging them together. They compared this ensemble model against all three models individually as well as many machine and deep learning models and found that these transformer models achieved the highest precision, accuracy, recall, and f1-score with the ensemble model achieving the absolute best score for all four of those measures. (Gundapu and Mamidi, 2021).

Another study (Rai et al., 2022) connected the output layer of a BERT model to an LSTM model. The BERT model is better at learning contextual relationships between words, which makes it easier for the LSTM to learn sentence semantics. They used training data from Politifact (a political news dataset), and GossipCop (a celebrity news dataset). Our training data focuses mainly on classification of political news data. The BERT + LSTM model outperformed all other models, including BERT alone, in all possible metrics. However, our study is primarily using the DistillBert transformer model as it is a lighter and faster model with a 95% retention of Bert's performance.

DistillBERT is a smaller, faster version of BERT. BERT stands for Bidirectional Encoder Representations from Transformers. BERT uses a masked language model (MLM) pre-training objective which randomly masks some of the input tokens and BERT's job is to predict the vocabulary id of the masked word based on its context. It is advantageous because it allows for context training based on both directions (words before and after), unlike others which usually only use words that come before. Because of this, it can be fine-tuned with only one additional output layer (Devlin et al., 2019). BERT outperforms most other machine learning methods in its ability to classify fake news (Rai et al., 2022). DistillBERT is a model that trained to reproduce the behavior of BERT, and is about 40 percent the size of BERT, allowing it to be 60 percent faster. However, it still retains about 97 percent of BERT's performance. Both of these models use the same architecture, but DistillBERT has about half the number of layers (Sanh et al., 2019). We attached a pre-trained DistillBERT model to a neural network and trained it on our training dataset to calculate its f1-score based on the test and YAKE data. The f1-score is a balanced metric of the precision and the recall of its performance on the test data. We found that the f1-score for the test data was very similar to the f1-score for the YAKE phrases we generated, meaning the phrases we generated are phrases commonly used in the titles of fake news articles.

YAKE (Yet Another Keyword Extraction) is an open source python-based keyphrase extraction that uses an lightweight, unsupervised learning approach to extract key phrases in the text. It can also extract keyphrases from different languages and is roughly equal in performance in other supervised keyword extraction model (Campos et al., 2018a). When the program is run, it calculates an list of tuples of a key phrase and it's score. The lower the score, the more relevant the key phrase is. It can also do generate both single and multiple words phrases, although for the purpose of the study, we are only working with phrases of length two and greater and testing their accuracy with a max key

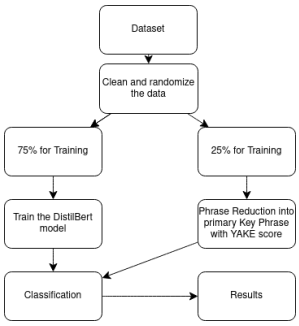

Figure 1: Model of our Methodology

word limit of twenty.

3 Methodology

Our study uses an combination of unsupervised and a supervised models. We primarily used DistilBert for the classification prediction of the key phrases generated with YAKE algorithm as shown in 1. We did not collect our data but instead used an existing dataset called WELFake for Fake News Classification from Kaggle that contained approximately 71,000 articles. The website does have list the wrong label legend as 0 represents Authentic articles and 1 for fake news. There was three main columns we used in this study, title, text, and label. The dataset also contained empty bodies and multiple languages; but the empty bodies was filtered out to prevent data skewing and there was a 7:1000 ratio of titles in a foreign language, so it is negligible to the final results. Then we split each dataset alongside it's label with approximately a 75-25 split and then combined to into our test and training datatset.

Alongside the classifier, we used YAKE to generate the key phrases from the testing data so it can be passed into the DistilBert model. Within YAKE, the KeywordExtractor function used to generate the keyphrases are set with the parameter of max ngram size of 20, window size of 20, and with a single key phrase to be generated. One of YAKE's features is that it can also generate single word key phrases but we disabled it as we were looking to generate phrases of length two and more. These parameters we set is so we can generate the most accurate phrases possible with YAKE, as with smaller parameters would generate only single word key phrases. There was also an issue where the title text may not be long enough to process into a multi-word key phrase, so to resolve this we filtered out that data point from the dataset to not be considered.

To run the DistillBERT model, we first tokenized the titles of all the articles so we could convert them into tensors, which DistillBERT could use to train on. Since the labels of all the articles were already '0'(real) or '1' (fake), all we had to do to get the labels was to map them to the real numbers 0, and 1 respectively. To account for the imbalance in the training data (not having an equal number of real and fake news articles), we weighted the classes (0 and 1) accordingly, based on how frequent they were, which we also converted to a tensor.

After that, we created our model, which was a neural network whose input data was first put through our already pre-trained DistillBERT model. We decided the max input for an article title should be 20 words, or 20 features, and every title less than 20 words long would be padded to 20 features and every title with more than 20 words would be truncated to 20 features. This is because for a neural network, each layer has a fixed number of inputs. We only used decided to only use about 5 hidden layers, because we found that using 10 hidden layers took quite a bit of time and during each epoch, the accuracy and loss function value didn't improve at all. After we passed our data through the hidden layers, we needed a flatten layer to reduce the data from the other layers into a tensor, which could be fed into the output layer (**?**). The output layer, which consisted of two nodes, which represented the two classes, "real" and "fake" (0 and 1). Finally, we used a soft-max layer to calculate all the probabilities of each data point being real or fake news (Saxena).

We then begun training our data. We used only 10 epochs, but originally started with 20 epochs. However, we found that the average value of the loss function for our testing data was increasing in the last ten epochs, meaning we were over fitting the data. For each data point, the model predicted the value of the training data, and then the loss function was used, taking into account the amount of real news articles as opposed to fake news articles, to calculate the loss value. Then, we then back-propagated, so we could update each of our weights based on the loss (**?**). After we were done training, we tested our model, for each epoch, on our training data and our testing data, so we could see how well it did on both. We calculated the accuracy and the average loss.

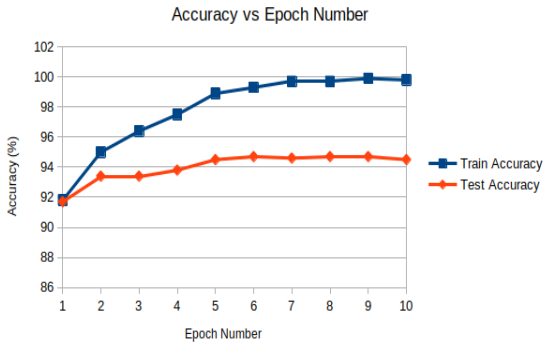

Figure 2: Test and Training Accuracy over all ten epochs

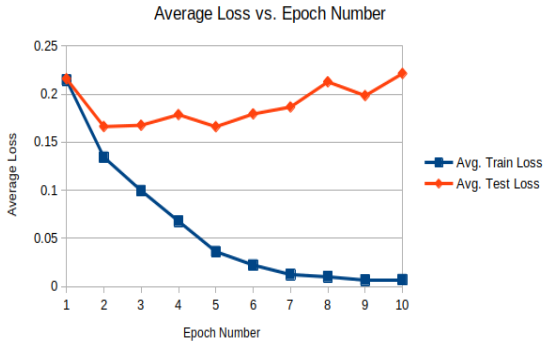

Figure 3: Average Test and Training Loss over all ten epochs

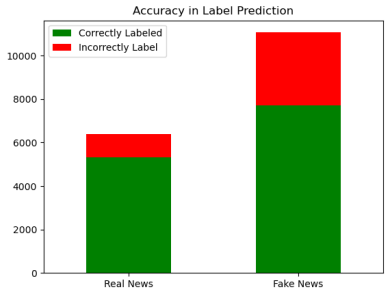

Figure 4: Uneven Number of Authentic vs Fake News

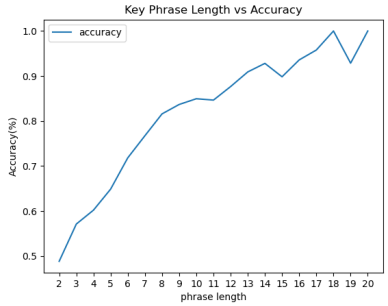

Figure 5: Accuracy increases with Phrase length

4 Results and Discussion

Within our DistillBERT model, our training accuracy started out quite high, over 90%, and kept on increasing and our training loss continued to decrease, as shown in figures 1 and 2. However, the average test loss began decreasing, and at around epoch 4, it hit its minimum value and started increasing again.

Unfortunately, the model didn't do so well on our YAKE generated phrases, which we ran only after all our epochs, with only a 74.5% accuracy and a 1.322549 average loss, which is insanely high.

We found that the f1-scores for both the testing data and our YAKE phrases, were not very high.

	Test	Yake
f1-score	0.64636	0.61173

So it appears our DistillBERT doesn't isn't actually very promising as a combination of all metrics, while the accuracy was very high.

It is clear that we needed to either decrease the number of epochs even more, and increase the number of layers (but due to timing constraints that wasn't an option). Just increasing the number of layers in the model without decreasing the number of epochs was already giving us unhelpful results, seeing as the accuracy and loss didn't change much at all. Our f1-scores were about the same as when we trained with one less layer, a lower max title length, and 20 epochs, so it is likely that the amount of epochs it takes to train alone does not affect the f1-score very much, even though it definitely can allow for overfitting of the data.

There are some values of the amount of epochs, the max title length, and the number of hidden layers that would have led to a very high accuracy and f1-score as well as a very low average loss on the final epoch, that we didn't have time to find. With a bit more time, we could have experimented more and tried to find them on a small scale, and then seen if we could've made them bigger.

Our testing dataset was also reduced drastically from the initial 20,000 data points to only about 17,000 data points as we removed noise from the testing data set. This may have affected in our results as there is only 6,405 total data points for authentic news titles compared to the 11,074 total data points for fake news titles as shown in 4. However, it is evident that within fake news, it is more prone to be misclassified incorrectly.

But how is it depended based on the phrase

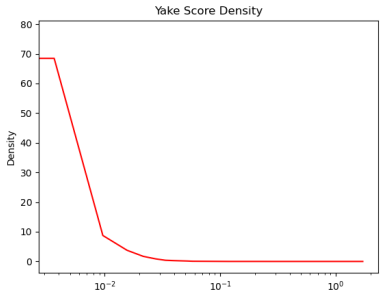

Figure 6: Most scores are significally small

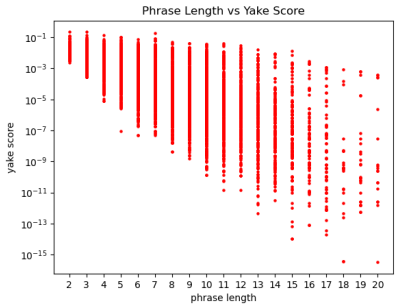

Figure 7: Downwards relationship

length? As expected, as we increase the amount of words in the Key Phrases, the accuracy of it's classification also increases. However we also noticed that even at only 2 words in the key phrase, the accuracy was already at 50% as shown in 5. At a word phrase length of 11, the accuracy reaches 84%; but we should note that this is close to the average length of the titles and that the classification of a false positive or negative may be due to the accuracy of classification of DistilBert model 2 and not due to length of the phrase itself.

But was the generated phrases actually representative to the actual key phrases? Both 6 and 5 are on a logarithmic scale, where the YAKE score of 0 represents the confidence level of 100%. Overall, bigger phrases tend to represent a more accurate score. But as shown in 6, most of the data points had a score smaller than 10^{-2}, with an average score being approximately 10^{-6} . This means that the phrases generated are extremely accurate to be the key phrases but with the limitations from the from earlier.

5 Conclusion

The results do show that we can accurately create key phrases from titles from YAKE. But we cannot be certain that key phrases from titles are a good indicator of its authenticity. It performed worse than just with the full testing data and only increased in accuracy as we increased the number of words in the phrase itself. If we were to run this project again, we would want to do a couple of things differently. Firstly, adjust the epochs to improve the accuracy of the DistilBert model. Secondly, I also want to increase the number data being test by implementing the text of the articles itself to examine each key phrase in each sentence. And lastly, I want to take the different length key phrases from each title to see if we get a higher accuracy from less words. But our results are inconclusive based on the current data we have.

6 References

References

Monther Aldwairi and Ali Alwahedi. 2018. Detecting fake news in social media networks. pages 215–222. Procedia Computer Science.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018a. A text feature based automatic keyword extraction method for single documents. In *Advances in Information Retrieval*, pages 684–691, Cham. Springer International Publishing.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018b. Yake! collection-independent automatic keyword extractor. In *Advances in Information Retrieval*, pages 806–810, Cham. Springer International Publishing.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Anshika Choudhary and Anuja Arora. 2021. Linguistic feature based learning model for fake news detection and classification. *Expert Systems with Applications*, 169:114171.

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

Sunil Gundapu and Radhika Mamidi. 2021. Transformer based automatic COVID-19 fake news detection system. *CoRR*, abs/2101.00180.

Suhail Yousaf Muhammad Ovais Ahmad Iftikhar Ahmad, Muhammad Yousaf. 2020. Fake news detection using machine learning ensemble methods. *Complexity*, 2020.

Chun-Ming Lai, Mei-Hua Chen, Endah Kristiani, Vinod Kumar Verma, and Chao-Tung Yang. 2022. Fake news classification based on content level features. *Applied Sciences*, 12(3).

Nishant Rai, Deepika Kumar, Naman Kaushik, Chandan Raj, and Ahad Ali. 2022. Fake news classification using transformer based enhanced lstm and bert. *International Journal of Cognitive Computing in Engineering*, 3:98–105.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. ArXiv.org.

Shipra Saxena. Introduction to Softmax for Neural Network — analyticsvidhya.com. https://www.analyticsvidhya.com/blog/2021/04/introduction-to-softmax-for-neural-network/.

Lilapati Waikhom and Rajat Subhra Goswami. 2019. Fake news detection using machine learning.

H Sirafi Z Khanam, B N Alwasel and M Rashid. 2021. Fake news detection using machine learning approaches. *IOP Conference Series: Materials Science and Engineering*, 1099.