CS230

Detecting COVID-19 Fake News Using Deep Learning

Anmol Tukrel
anmol.tukrel@stanford.edu

Avalon Wolfe
avalonw@stanford.edu

Karissa Yau
kcyau@stanford.edu

Abstract

With COVID-19 emerging as a pandemic that has affected everyone worldwide, people have become more reliant on news to make everyday decisions to ensure their safety of themselves and their loved ones. However, fake news is almost becoming a "second pandemic" or "infodemic," posing as a health hazard to people worldwide. Given that coronavirus-related fake news is such a new phenomenon, prior work has not applied fake news detection to coronavirus.

In an effort to tackle this issue, we utilize a modified LSTM that considers features relevant to fake news including the Jaccard index between the title and text, polarity, and frequency of adjective use. Our model was trained on a 600 article dataset containing 300 fake news articles and 300 real news articles. It achieved an overall accuracy of 0.91 with F1 scores of 0.89 and 0.92 for real and fake news, respectively.

1 Introduction

The proliferation of fake news in the era of coronavirus is a particularly pressing issue. Whether it is the vandalism of cell towers in the UK after rumors about a link between 5G and coronavirus or the promotion of unproven medication, the spread of coronavirus-related fake news has profound ramifications. The United Nations reiterates this in writing that "a single falsehood that gains traction can negate the significance of a body of true facts," demonstrating how imperative it is to combat false information.

The primary way that social media platforms such as Facebook and Google have tackled this issue is through the use of independent fact-checkers around the world. It is clear, though, that this strategy is not sustainable in the long run. Fact checkers are often unable to keep up with vast troves of information that they have to sort through each day. Our project hopes to tackle this issue through the combination of a novel dataset and deep learning.

2 Related Work

With the rising importance of detecting fake news, various models have been used to distinguish between real news and fake news. The most common architectures used with fake news classification are CNNs, RNNs, and LSTMs. Many researchers build upon these architectures to produce more complex architectures, such as TraceMiner. TraceMiner is a sequence classifier built using LSTM-RNNs that is used to model the diffusion of a message on social media and, ultimately, uses softmax to produce a predicted class label. It aims to determine whether information is fake or not on social

CS230: Deep Learning, Spring 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

media by looking at how messages spread through a network, as this gives a strong indication of what type of information it contains.

Additionally, researchers have used a CNN with max pooling for fake news detection. This CNN starts off by extracting bigrams, trigrams, 4-grams and 5-grams from the text, which is fed into a linear layer and the max pooling layer selects one of the scaled inputs. The results are added and transformed via a ReLU unit, and finally the the output is determined using a sigmoid function. Other architectures include FAKEDETECTOR and DEEPWALK. FAKEDETECTOR uses a Gated Diffusive Unit that fuses different inputs for output generation with content "forget" and "adjust" gates. DEEPWALK uses network embedding to embed news articles and authors to a latent feature space.

3 Dataset

3.1 Dataset Features

Our dataset consists of coronavirus-related real news and fake news articles. It has approximately 300 real news articles and approximately 300 fake news articles. Real news articles were randomly selected out of the 52,000 articles in the Covid-19 Public Media Dataset from Anacode while fake news articles were manually collected. Each article includes information about the title, author, text and whether it is real (denoted by 0) or fake (denoted by 1). While "fake news" is a broad definition that encompasses many types of articles such as satire, political, false health and scientific news, and more, our dataset includes articles from various sources and topics to ensure a diverse dataset.

3.2 Data Collection

We manually collected approximately $1/3$ of our fake news from reputable fact-checking sites, primarily collecting articles deemed to be fake by Politifact, Poynter, Full Fact, Lead Stories, and the East StratCom Task Force's EUvsDisinfo project. When collecting from fact-checking services that rated articles on a spectrum from true to false, we only collected articles that were predominantly false (e.g. we collected from Politifact's 'False' and 'Pants on Fire' categories but not from 'Mostly False' or 'Half True').

We collected approximately $2/3$ of our dataset by combing through sites that are known to publish fake news and disinformation. We primarily looked at 88 websites from the United States that NewsGuard identified as "publishing materially false information about the virus," as well as several other sites that we found were notorious purveyors of fake news. We cross-referenced the facts mentioned in each news article about coronavirus against fact-checking sites and, when those weren't useful, against multiple reputable media sources (such as the BBC, the NYT, the WSJ, and the Economist) to ensure that there was a consensus that disproved the information in a fake news article.

We collected each news article in our coronavirus fake news dataset manually. We also read each individual article to ensure it was verifiably false, meaning that the overarching theme or facts stated in the article could be refuted by trustworthy news sources. Furthermore, we used sites such as snopes.com, politifact.com, or newsguardtech.com to find domains that were known for spreading misinformation.

4 Methods

4.1 Baselines

In order to evaluate our model, we compared our results to three baselines including: a naive bayes classifier, a three layer neural network, and an LSTM. For each of our models, we split up our collected dataset with 80% of the examples used for training and 20% used for testing. We chose to use the naive bayes classifier as one of our baselines as it is often used as a baseline when dealing with text classification. Our neural network also found moderate success (discussed further in the Results section) after being trained for 10 epochs. For our neural network baseline, we used a learning rate of 0.001, gradient descent optimization, and sparse softmax cross entropy for our loss function. Finally, we trained our baseline LSTM for 4 epochs with a batch size of 25 and learning rate of 0.01. With

our baseline LSTM, we also used a sigmoid activation function, binary cross entropy for our loss function, and Adam optimization.

4.2 Model Architecture

We chose to build upon the baseline LSTM as it was the best performing baseline model. When exploring additional features to add, we considered the structure and style in which fake news about Covid-19 was written. Similar to other forms of fake news, we found that Covid-19 fake news is written using highly opinionated and polarizing language. In contrast, real news uses a much more neutral tone. We also noticed that fake news sources uses many more adjectives than real news articles. Finally, we noticed that fake news articles are much more likely to include clickbait titles.

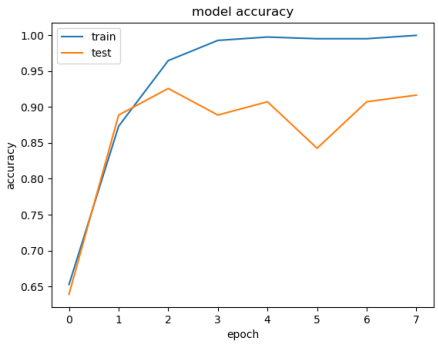
Thus, we experimented with including the following input features alongside the word embeddings of the body text:

- The number of adjectives present in the body of the article
- The Jaccard index between the title and body text
- Subjectivity of the body text (i.e. the degree to which the author's language referred to personal opinion, emotion or judgment)
- Polarity of the body text

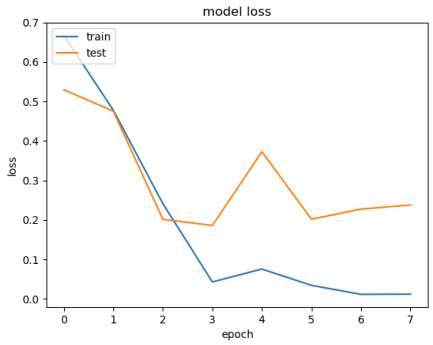
Upon experimenting with the addition of the aforementioned parameters, our model had the best performance when using the number of adjectives, Jaccard similarity of the title and body text, and polarity of the body text as additional parameters to the LSTM.

For our LSTM, we chose hyperparameters simply by testing many different combinations of values and choosing the ones with the best F1 scores. We experimented with batch sizes ranging from 8 to 64, learning rates ranging from 0.001 to 0.05, and number of epochs ranging from 3 to 50. After experimentation, we used a batch size of 25 and learning rate of 0.01 for our final model. Finally, we limited the number of epochs to 8 since this is when we noticed the loss flattening out.

Additionally, we chose to use Adam optimization, a sigmoid activation function, and a binary cross entropy loss function as our task involved binary classification.



Accuracy vs Number of Epochs



Loss vs Number of Epochs

5 Results and Discussion

The following figures contain the results for our a modified LSTM that considers features relevant to fake news including the Jaccard index between the title and text, polarity, and frequency of adjective use and compares it to three baseline models.

Our model was trained and tested on a 600 article dataset containing 300 fake news articles and 300 real news articles. The dataset was split into two parts with 80% of articles used for training and 20% used for testing. Our model achieved an overall accuracy of 0.91 with F1 scores of 0.89 and 0.92 for real and fake news, respectively. This was marginally better than the LSTM, which was the

best performing baseline model with F1 scores of 0.89 and 0.90 for real and fake news, respectively. Surprisingly, our Naive Bayes baseline model performed better than our neural network baseline. We believe the Naive Bayes baseline performed better than the neural network due to the small size of our dataset, and the neural network not being able to learn as well from the small dataset.

Initially, when we started this project, we built an RNN that trained on an imbalanced dataset with approximately 7000 real news articles and 300 fakes news articles. This model achieved an accuracy score of 0.6 with F1 scores of 0.95 and 0.27 for real and fake news, respectively. Our other baseline models mentioned previously had similar unsatisfactory F1 scores and confusion matrices. As a result, we decided to create a balanced dataset with half of the articles being real coronavirus news

	precision	recall	f1-score	support
0	0.83	0.89	0.86	45
1	0.92	0.87	0.89	63
accuracy			0.88	108
macro avg	0.88	0.88	0.88	108
weighted avg	0.88	0.88	0.88	108

(a) Naive Bayes Classifier Results

	precision	recall	f1-score	support
0	0.82	0.82	0.82	45
1	0.87	0.87	0.87	63
accuracy			0.85	108
macro avg	0.85	0.85	0.85	108
weighted avg	0.85	0.85	0.85	108

(b) Neural Network Results

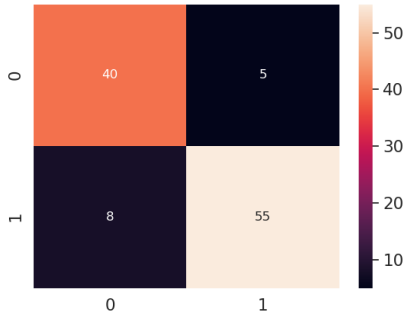
	precision	recall	f1-score	support
0	0.85	0.94	0.89	48
1	0.95	0.87	0.90	60
accuracy			0.90	108
macro avg	0.90	0.90	0.90	108
weighted avg	0.90	0.90	0.90	108

(c) LSTM Results Results

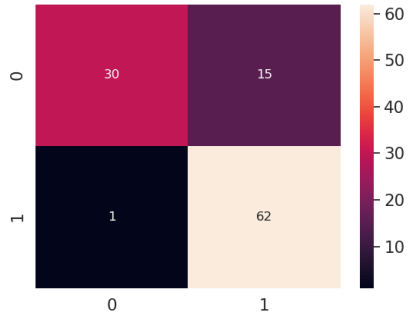
	precision	recall	f1-score	support
0	0.89	0.89	0.89	45
1	0.92	0.92	0.92	63
accuracy			0.91	108
macro avg	0.90	0.90	0.90	108
weighted avg	0.91	0.91	0.91	108

(d) LSTM with Additional Features Results

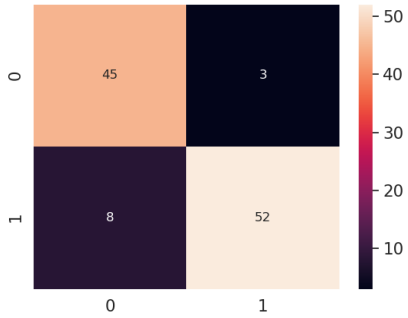
Figure 1: Results for the three baselines and our model



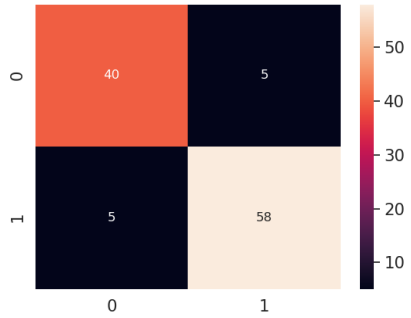
(a) Naive Bayes Classifier Confusion Matrix



(b) Neural Network Confusion Matrix



(c) LSTM Confusion Matrix



(d) LSTM with Additional Features Confusion Matrix

Figure 2: Confusion matrices for three baselines and our model

and the other half being fake coronavirus news. Due to training on the balanced dataset, we achieved significantly better results, as seen by the figures above.

With regard to our modified LSTM, using the additional input features to the LSTM increased accuracy slightly, by merely 0.01. Evidently, the use of the additional features outlined in section 4.2 did not have much of an impact on our model's ability to detect fake coronavirus news. We believe that this may be caused by the small number of examples for the model to learn from or because many of these characteristics (such as polarity and number of adjectives) of the input text may be learned by the baseline LSTM without having to explicitly calculate and provide them as inputs to the model. We believe that training on a larger dataset, and considering additional features such as punctuation, number of quotes, or ratio of factual statements to opinion statements may help to improve our model's performance.

6 Conclusion and Future Work

In this project, we demonstrated the ability for deep learning to help address fake news regarding coronavirus, one of the key challenges governments face in battling the Covid-19 pandemic. By using a modified LSTM that trains on features unique to fake news, our project achieved an overall accuracy of 0.91 with F1 scores of 0.89 and 0.92 for real and fake news, respectively.

One area for future improvement is to extend the model to distinguish between different types of fake news. The term fake news is broad and often encompasses a variety of information such disinformation, misinformation, hoaxes, propaganda, satire, rumors, and click-bait. A satirical article is very different from an article disinformation, demonstrating how wide of a spectrum is included within fake news. In future work on this model, we would be interested in classifying between real news and the different categories of fake news. This is something that would require altering our dataset but will likely work with our existing model given some fine-tuning.

Another area for potential improvement is to extend the model to classify between coronavirus-related real news and fake news on social media. The majority of fake news is disseminated through social media, particularly Facebook and Twitter. And, though they attempt to remove fake news, social media platforms are often unable to flag all posts. This is saliently identified by researchers who, with a small sample set of 649 posts on Facebook and Twitter, found that 90% of posts remained online without any warnings. This is particularly concerning since fake news on social media can be posted by anyone who has an account, and it has the opportunity to reach a large audience through shares or retweets. Moreover, one of the difficulties in identifying fake news on social media is that it is often conveyed through memes. Since memes include text in a picture format rather than a text-based caption, it is much more challenging to use text classification. In the future, we would be interested to feed text-based social media posts into our model and implement a text recognition algorithm to identify text in memes.

With the proliferation of coronavirus-related fake news online, it is clear that this is a critical issue to solve in order to control the Covid-19 pandemic. Our project aims to tackle this "second pandemic" through the use of a modified LSTM with features that attempt to capture the unique attributes of fake news. While there are many areas to extend upon such as detecting fake news on social media, our project creates a novel dataset and applies deep learning to solve an issue at the center of the Covid-19 pandemic.

7 Contributions

Anmol Tukrel: Implemented naive bayes and neural network baseline, researched possible additional features, experimented with various hyperparameters, Google Cloud setup, helped write script to collect additional features, helped to implement modified LSTM, and worked on final report.

Avalon Wolfe: Fake news and real news data collection, implemented LSTM baseline, researched possible additional features, experimented with various hyperparameters, helped to implement modified LSTM, and worked on final report.

Karissa Yau: Fake news data collection, wrote a script to collect the additional features such as polarity, subjectivity, Jaccard score, and adjective count, and created the video presentation.

8 Acknowledgements

We would like to thank our CS 230 TA, Jonathan Li, for his guidance and support this quarter!

References

[1] Sholk, Gilda. (2017) "Evaluating Machine Learning Algorithms for Fake News Detection." *110-115. 10.1109/SCORED.2017.8305411.*

[2] Bajaj, Samir. (2017) "Fake News Detection Using Deep Learning". *https://pdfs.semanticscholar.org/19ed/b6aa318d70cd727b3cdb006a782556ba657a.pdf*

[3] Balwant, Manoj Kumar. (2019) "Bidirectional LSTM Based on POS tags and CNN Architecture for Fake News Detection." IEEE. DOI: 10.1109/ICCCNT45670.2019.8944460.

[4] Bouronje, Peter et al. (2017) "From Clickbait to Fake News Detection: An Approach based on Detecting the Stance of Headlines to Articles." *https://www.aclweb.org/anthology/W17-4215.pdf.*

[5] Thota, Aswini. (2018) "Fake News Detection: A Deep Learning Approach." SMU Data Science Review: Vol. 1 : No. 3 , Article 10. *SMU Data Science Review: Vol. 1 : No. 3 , Article 10.*

[6] Yang, Yang. (2018) "TI-CNN: Convolutional Neural Networks for Fake News Detection." *arXiv:1806.00749 [cs.CL]*

[7] Shrestha, Anish. (2019) "Fake News Classification using Long Short Term Memory." *https://medium.com/@sthacruz/fake-news-classification-using-glove-and-long-short-term-memory-lstm-a48f1dd605ab*

[8] Zhang, Jiawei et al. (2018) "FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network." *https://arxiv.org/pdf/1805.08751.pdf*

[9] Rodriguez, Alvaro and Lara Iglesias. (2019) "Fake News Detection Using Deep Learning." *https://arxiv.org/pdf/1910.03496.pdf*

[10] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova Kristina. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." https://arxiv.org/pdf/1810.04805.pdf

[11] Pierri, Francesco and Stefano Ceri. (2020) "False News On Social Media: A Data-Driven Survey." *https://arxiv.org/pdf/1902.07539.pdf*

[12] BBC News. "Social media firms fail to act on Covid-19 fake news." *https://www.bbc.com/news/technology-52903680*

[13] Fake News Detection, (2018), GitHub Repository, https://github.com/rockash/Fake-news-Detection