CS230

COVID-19 Misinformation Classification with Convolutional Neural Networks

(Category: Natural Language Processing)

Stephanie X. Hu
Department of Computer Science
Stanford University
stephhu1@stanford.edu

Abstract

Within recent years, misinformation has proliferated across various social media platforms. During the COVID-19 pandemic, in particular, such misinformation has counteracted progress towards the protection of public health. To address the rising amounts of COVID-19 related misinformation online, I have used a Convolutional Neural Network (CNN) to classify the content of tweets represented with pre-trained word embeddings as true or false. The ultimate test accuracy achieved was 94.02%.

1 Introduction

Misinformation online can stand as a significant obstacle to mass recovery from the COVID-19 pandemic, as it spurs vaccine hesitancy, the use of unproven treatments, and a lack of adherence to social distancing or mask-wearing mandates. In turn, I have input the text of COVID-19 related Twitter posts represented with pre-trained word embeddings [7] and used a Convolutional Neural Network (CNN) to output predictions of whether the stated claims are true or false.

Notably, CNNs have shown promising performance on recent text classification tasks [5][6][9], despite their predominant applications in image classification. Similar to pixels that traditionally constitute an image passed to a CNN, the aforementioned word embeddings can quantitatively represent text sequences. CNNs can then be trained to recognize patterns from such text representations, such as local and position-invariant key phrases [6] that can be critical for misinformation classification.

Social media platforms, in particular, have served as major spaces in which misinformation is circulated [4]. In turn, I wish to continue to broaden the range of research of CNNs, specifically within the realm of natural language processing, onto a novel subject of COVID-19 misinformation classification in regards to social media. Ultimately, the algorithm may then be used to aid public health measures by allowing professionals to pinpoint and subsequently correct such misinformation during the current crisis.

2 Related Work

Traditional technical methods for text classification tasks center on Logistic Regression and Naive Bayes classifiers. Researchers have previously used such classifiers to detect misinformation in fake news articles [3] and tweets [1]. A primary strength of Logistic Regression and Naive Bayes classifiers is in their relative speed, particularly compared to more complex algorithms. However,

CS230: Deep Learning, Spring 2021, Stanford University, CA.

since both rely on linear decision boundaries, they may not be able to achieve a more optimal accuracy that a neural network is capable of with more complex decision boundaries.

Another approach for online misinformation classification is a LSTM-RNN. In particular, prior research has classified tweets as real or fake news with a LSTM-RNN through the traces of a message [8]. Traces of a message have been defined as the data capturing when the message was posted or shared as well as the source that posted or shared it. In turn, the LSTM-RNN utilizing a message's traces takes advantage of how misinformation can be spread from similar sources and at similar times. However, since the aforementioned LSTM-RNN does not consider the text of social media posts, it does not take in account key phrases within such text that may be critical for classifying misinformation from a source that does not typically share fake news.

Lastly, CNNs have also been used for text classification tasks. As previously mentioned, CNNs have recently shown notable success in recognizing patterns from text, such as local and position-invariant key phrases [6]. Past studies have used CNNs for tweet sentiment classification [5] and the classification of news stories by subject (e.g. sports) [9]. To add to the studies, I have similarly used a CNN for my text classification task. However, unlike the referenced literature, I will consider COVID-19 related misinformation specifically rather than a broader category of text, and will classify a text's truth or falsity rather than its sentiment or subject.

3 Dataset

The dataset [2] consists of 10700 COVID-19 related posts collected on Twitter. Each entry contains the text of a tweet and a label that signifies whether the content is real or fake news. Moreover, the data is relatively balanced: 5100 tweets are classified as misinformation while 5600 tweets are classified as factual text. The training set, validation set, and test set is composed of 60%, 20%, and 20% of the dataset, respectively. Such a split corresponds to 6420 tweets reserved for the training set, 2140 tweets for the validation set, and 2140 tweets for the test set.

Tweet	**Label**
Protect yourself and others from #COVID19 when using public transportation. Practice social distancing avoid touching surfaces and practice hand hygiene. Learn more: https://t.co/0vhHD4uFv9. https://t.co/D8YSeE3vXv	real
Mixing aspirin paracetamol honey and lemon cures COVID-19.	fake

Table 1: Samples from the COVID-19 Fake News dataset.

Prior to training, the data was preprocessed to replace all URLs with a designated `url` token, remove non-alphabetic characters, remove single characters, convert all letters to lowercase, and truncate or pad all tweets to a constant tweet length of 40 tokens. I then used 200-dimensional word embeddings pre-trained by a GloVe model on a Twitter corpus [7] to create representations of each tweet input for the algorithm.

4 Methods

To address the text classification task, I have implemented two deep learning architectures: a simple baseline neural network and a CNN.

Baseline Neural Network

For the simple baseline neural network, 40-dimensional inputs are passed to an embedding layer and subsequently flattened before progressing to a dense layer with a ReLU activation function, and then another dense layer with a sigmoid activation function. The weights of the embedding layer are provided with an embedding matrix created through stacking the pre-trained word embeddings for distinct words within the COVID-19 tweets of the training set.

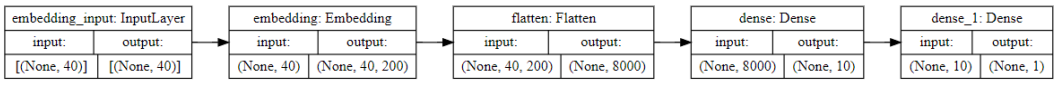

Figure 1: Baseline neural network model architecture.

Convolutional Neural Network

The CNN uses 40-dimensional inputs that are passed to an embedding layer similar to the baseline neural network before progressing to a one-dimensional convolutional layer with 128 filters, a kernel size of 3, and a ReLU activation function; a global max pooling layer; a dropout layer with a dropout rate of 0.2; a dense layer with a ReLU activation function; and a dense layer with a sigmoid activation function.

Both the baseline neural network and CNN use the Adam optimizer ($\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$) with binary cross-entropy loss

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=0}^{N} y_i log(\hat{y_i}) + (1 - y_i) log(1 - \hat{y_i}).$$

Additionally, both use a batch size of 32 and train for 6 epochs.

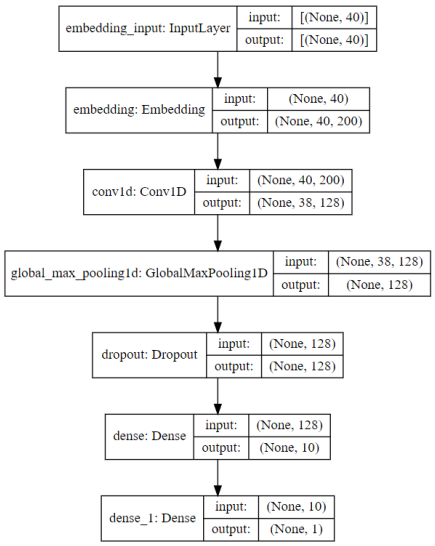

Figure 2: CNN model architecture.

5 Experiments/Results/Discussion

To reach the aforementioned CNN model, I tuned several hyperparameters, including the number of epochs, batch size, number of filters and kernel size in the convolutional layer, and dropout rate.

To maximize true positives and true negatives within a relatively balanced dataset, I used accuracy as the optimizing metric. Moreover, to additionally consider false positives and false negatives in the misinformation classification task, I used the F1 score and AUC as satisficing metrics that should at least meet the baseline neural network's performance on the test set. The baseline neural network achieved a test accuracy of 0.8724, F1 score of 0.8787, and AUC of 0.943 (see Fig. 4 and Table 2).

Considering the optimizing metric, the number of epochs was chosen after training with 100 epochs and observing that training accuracy reaches a plateau and validation accuracy peaks around 6 epochs (see Fig. 3). The training loss also starts to reach a trough alongside a rise in validation loss around 6 epochs. The batch size of 32 was chosen after testing an array of batch sizes between 16 and 256 and pinpointing the batch size corresponding to the maximum validation accuracy.

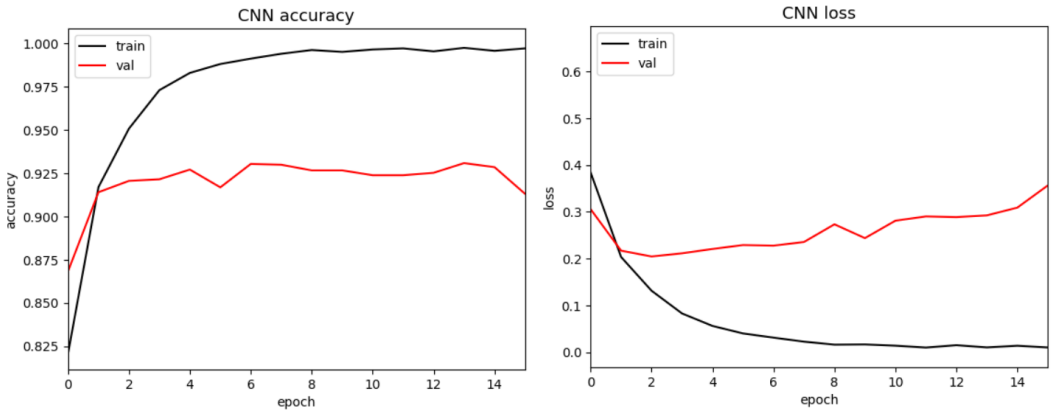

Figure 3: Plots of accuracy and loss over epochs for the CNN model.

To tune the hyperparameters of the convolutional layer, I used multiple combinations of the number of filters and kernel size in the layer. Notably, while keeping the number of filters constant at 64, raising the kernel size from 3 to 5 resulted in a drop in validation accuracy from 0.9182 to 0.9061, and a drop in the F1 score from 0.9221 to 0.9106. While keeping a kernel size of 3 constant, raising the number of filters from 64 to 128 resulted in a slight rise in validation accuracy from 0.9182 to 0.9220 and a slight rise in the F1 score from 0.9221 to 0.9262.

I had also observed that the CNN without a dropout layer had overfit the training data with a training accuracy of 1.0000 and validation accuracy of 0.9220. To mitigate the overfitting, I later added a

dropout layer. To tune the dropout rate, I used one constant convolutional layer with the number of filters and kernel size set at 128 and 3, respectively. A model with a dropout rate of 0.2 led to the highest validation accuracy of 0.9304 and F1 score of 0.9335. In contrast, a dropout rate of 0.5 resulted in a validation accuracy of 0.9182 and F1 score of 0.9202.

Along with the aforementioned changes, I tried adding a second convolutional layer. However, even with a dropout layer, the added convolutional layer resulted in more overfitting and thus a lower validation accuracy than a single convolutional layer. Compared to the model with one convolutional layer containing 128 filters and a kernel size of 3, a model composed of two of such layers and a dropout rate of 0.2 resulted in a drop in validation accuracy from 0.9304 to 0.9182 and a drop in the F1 score from 0.9335 to 0.9217. A similar model with the two convolutional layers but a dropout rate of 0.5 resulted in a validation accuracy of 0.9093 and F1 score of 0.9129.

A visualization of the AUC metric through ROC curves for the models tested with varying numbers of filters, kernel sizes, and dropout rates are pictured in Fig. 4. The model with the largest AUC of 0.984 is the CNN with one convolutional layer, 128 filters, a kernel size of 3, and dropout rate of 0.2.

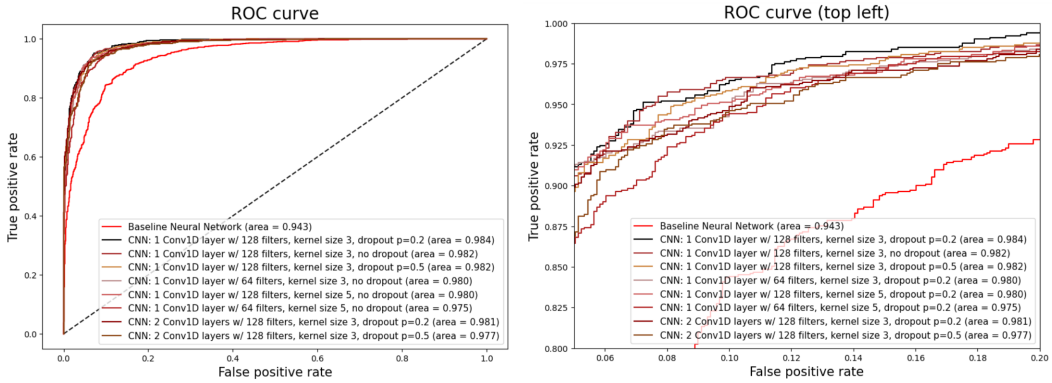

Figure 4: ROC curves for varying hyperparameters on the test set.

Architecture	Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1
Baseline	0.9988	0.8724	0.8815	0.8760	0.8787
CNN	0.9891	0.9402	0.9364	0.9513	0.9438

Table 2: Performance metrics for the baseline neural network and the CNN model with filters = 128, kernel_size = 3, and rate = 0.2.

After analyzing the aforementioned combinations of hyperparameters through accuracy, the F1 score, and AUC, we can see that the most optimal combination for the CNN model was 6 epochs, a batch size of 32, a single convolutional layer containing 128 filters and a kernel size of 3, and a dropout rate of 0.2. The CNN model with those hyperparameters ultimately achieved an accuracy of 0.9402 and F1 score of 0.9438 on the test set, which can be compared to the baseline neural network accuracy of 0.8724 and F1 score of 0.8787 on the test set (see Table 2).

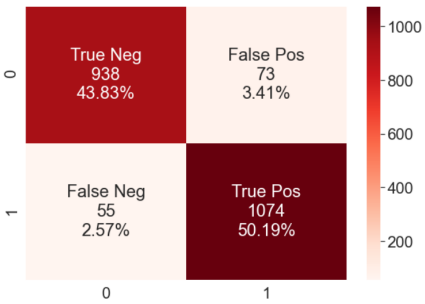

Figure 5: Heatmap for tuned CNN model on test set.

As seen in Fig. 5, the tuned CNN model produced 1074 true positives, 938 true negatives, 73 false positives, and 55 false negatives on the test set, where a positive classification is real news while a negative classification is fake news. The strong classification performance of the CNN model, particularly relative to the baseline neural network, is likely due to the algorithm's capability to recognize patterns like local and position-invariant key phrases [6] that can differentiate misinformation from real news. As an example, the phrase "Bill Gates," which occurs a total of 89 times in the dataset and only in tweets labeled as fake, can potentially be pinpointed as one of such critical key phrases.

Misclassified Tweet	Predicted Label	True Label
The government raided The Medical City and other private hospitals taking away their personal protective equipment (PPE).	real	fake
Fatality rates among those infected by COVID-19 range up to 0.054% in those aged 70 and over.	real	fake
Police have shut down a series of illegal parties overnight as people enjoyed a final weekend of revelry before tougher coronavirus restrictions come into force https://t.co/bxV22ZH2Ts	fake	real
Coronavirus: Dogs deployed at Helsinki Airport to sniff out virus https://t.co/BF2elPeTh3	fake	real

Table 3: Samples of misclassified tweets from the COVID-19 Fake News dataset.

However, a commonality that was apparent among many of the false positives and false negatives is the presence of potential key phrases in the misclassified tweets associated with the opposite of the true labels. For example, the phrases "personal protective equipment" and "fatality" primarily occur in tweets with real news in the dataset; tweets with fake news containing such phrases would often be misclassified with a real label (see Table 3). Similarly, "police", "shut down", "illegal", and "dogs" primarily occur in tweets with fake news in the dataset, and tweets with real news containing such phrases would often be misclassified with a fake label.

6 Conclusion and Future Work

In this study, I developed a Convolutional Neural Network (CNN) to classify the content of COVID-19 related posts on Twitter as real or fake. The highest-performing CNN model achieved an accuracy of 94.02% on the test set, which can likely be attributed to a CNN's ability to recognize local key phrases that suggest a tweet's truth or falsity regardless of their position in the tweet. In turn, although CNNs are traditionally used to classify images, the CNN model has shown similarly strong performance on a misinformation classification task while using word embeddings to represent text.

For future work, I would collect more data from other social media platforms to generalize the CNN model to COVID-19 related posts beyond those on Twitter, and adjust my model accordingly. In particular, such platforms could include Facebook or Instagram. Similarly, I could also add COVID-19 related news headlines to the dataset alongside the social media posts. By broadening the scope of the data, the model can potentially be more accurate in pinpointing misinformation that is not necessarily in the format of a tweet. Moreover, the added data may also heighten the model's accuracy when classifying tweets as well. More specifically, we have discussed how the misclassified tweets in Table 3 contain phrases that primarily occur in the class opposite to the true label. In turn, the new data from other social media platforms or news headlines may provide more balance to where such phrases appear. For instance, there may be more Facebook posts with fake content that contain the phrase "personal protective equipment", which currently primarily occurs in tweets with real news in the dataset. The additional data from other social media platforms and news headlines could then potentially reduce the false positives and false negatives that were seen on the current test set.

7 Contributions

I worked alone on my project, but would like to thank my CS230 Project TA Prerna Khullar for her guidance throughout the quarter.

References

[1] Aborisade, O., and Anwar, M. (2018, July). Classification for authorship of tweets by comparing logistic regression and naive bayes classifiers. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)* (pp. 269-276). IEEE.

[2] Aghammadzada, E. (2021, March). COVID19 Fake News Dataset NLP, Version 1. Retrieved April 15, 2021 from https://www.kaggle.com/elvinagammed/covid19-fake-news-dataset-nlp.

[3] Al Asaad, B., and Erascu, M. (2018, September). A tool for fake news detection. In *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)* (pp. 379-386). IEEE.

[4] Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., ... & Zittrain, J. L. (2018). The science of fake news. *Science, 359*(6380), 1094-1096.

[5] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188.*

[6] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep Learning–based Text Classification: A Comprehensive Review. *ACM Computing Surveys (CSUR), 54*(3), 1-40.

[7] Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation.

[8] Wu, L., & Liu, H. (2018, February). Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the eleventh ACM international conference on Web Search and Data Mining* (pp. 637-645).

[9] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *arXiv preprint arXiv:1509.01626*.