Classification of Fake and Real News from Filipino Sources

Using Dense, LSTM and Convolutional Neural Networks

Submitted to:
Felix Muga II PhD

Submitted by:

Benjamin Ang

Alexander Pino Jr.

Dane Rosario

Jeremy Tan

I. Introduction

Driven by increasing Internet-based consumption of media, a "post-truth" society has emerged in which people prefer information which aligns to their pre-existing beliefs, rather than seeking out the true nature of a situation. This was precipitated, and continues to be worsened, by the emergence of "fake news". Comprising inadvertently-spread misinformation and deliberately-spread disinformation, fake news has been rampant globally (De Paor and Heravi, 2020), and the Philippines is no exception. According to a study by Pulse Asia conducted in September 2022, 9 out of 10 Filipinos (about 86 percent) see “fake news” as a pressing issue in the country (Lalu, 2022). This survey was conducted by doing face-to-face interviews with 1200 correspondents with a 2.8 margin of error and a 95% confidence interval.

It was found in the same study that 68% of the population have mostly interacted with fake news on the internet. Social media, in particular, has changed the public's conception of news, enabling citizen journalism to challenge the trust of established media on an open platform, while amplifying the "bandwagon effect" through popularity ratings and the attachment of familiar names in "shared" posts. This effect is now known to be exploitable with the use of automated and fake accounts (Tandoc Jr., 2017), further emphasizing the need to have an objective source which can inform readers of the veracity of a certain article. A neural network-based algorithm may be trained to perform this task in lieu of human moderators who may be prone to bias and unable to match the pace of social networks. This will be the objective of this paper.

II. Dataset

The "Fake News Filipino" dataset developed by Cruz et al. (2020) will be used for model training, testing and validation. It comprises 1 603 each of real news articles, sourced from major news websites including Pilipino Star Ngayon, Abante and Bandera, and fake news articles sourced from websites tagged as such by the National Union of Journalists in the Philippines (NUJP) and the non-profit independent organization Verafiles. All articles are written in Filipino, although typically also including English vocabulary. We acknowledge several biases and systematic errors may have persisted in the creation of this dataset; regardless of the website in which an article was published, it may be vulnerable to external influence by government, advertisers and audience, and internal subjectivity from its author

(Tandoc Jr., 2017). Furthermore, NUJP, specifically, is an organization of Filipino journalists currently employed by news organizations, who may be biased in their classification of fake news websites (NUJP, n.d.).

Label	Article
0 (Real)	Ayon sa TheWrap.com, naghain ng kaso si Krupa, 35, noong Huwebes dahil nakaranas umano siya ng emotional distress bunga ng mga malisyosong pahayag ni.....
1 (Fake)	Nagbigay na rin ng opinyon ang mga mix martial artists at organizers sa bansa kaugnay sa kumalat na video ng batang atenista na nambubugbog ng kapwa…

Figure 1. Sample data from the “Fake News Filipino” dataset

The dataset, stored in a comma-separated value (CSV) file format, is separated into two columns; "article" contains the text of a certain news article, and "label" contains a boolean value of 0 with a real article and 1 with a fake article. Initial preprocessing is minimal and only included encoding into a common UTF-8 character format, without the correction of spelling, grammar and punctuation (Cruz et al., 2020). Three deep learning methods will be used to perform a binary classification prediction on the boolean value; a densely-connected neural network, a long short-term memory neural network, and a convolutional neural network.

III. Methodology

Preprocessing

The dataset will be preprocessed identically for all models to be fitted. 25% of the dataset will be separated into the testing dataset, which will be held out until the final evaluation of the models. Byte-pair encoding is used to tokenize the text, an algorithm which separates text into subword units by learning the internal structure of individual words. This allows a better representation of the Filipino language which is morphologically-rich, being diverse in terms of its use of prefixes, suffixes and infixes within words (Cruz et al., 2020, p. 2598). After tokenization and filtering to the most common tokens, one-hot encoding will be used to convert tokens into numerical data for calculations within the model. Several hyperparameters, including the maximum number of subword units and length of a sequence, will be tuned to achieve the best performing model with a reasonable use of resources. Based on the mean length of each article at 183 words, we set the max sequence limit at 200 subwords. The vocabulary size of each of our models was set at 20,000 subwords.

Dense Neural Network

For our simple flattened dense network, a sequential stack of layers format for the model was specified using the Keras module. First, we embedded the contents of each news article into a numerical 2-D vector with dimensions of the dimension. Then, we flattened the vectors for each instance into a one dimensional vector. Then, we defined two dense layers as well as their activation functions. Since we are dealing with a binary classification problem, our activation function should output a value between 0 and 1 depending on certain parameters. The output should correspond to the probability that the news article is fake or not. In this case, we used both the sigmoid and ReLU (rectified linear unit) activation functions.

Convolutional Neural Network

Compared to previous neural networks, Convolutional Neural Networks specializes in analyzing spatial location of words. To do so, we will use the same data from the previous methodologies with identical preprocessing and feature engineering recipes. Next, we create a simple CNN model by constructing the following layers using the keras package: embedding layer, single one-dimensional convolution layer, global max pooling layer, densely connected layer, and a dense layer with a sigmoid activation function, which would

aid us in binary classification.The model's dimensionality will be managed by selecting the appropriate number of layers and kernel size which would result in a positive length for the sequence.

Long Short-Term Memory (LSTM) Neural Network

LSTMs are neural networks specialized to handle long, sequentially-ordered data, where the task of predicting an outcome such as completing a sentence might rely on information far along the sequence. The Keras library will be used to implement a single LSTM layer between an embedding layer and a densely connected layer which will condense the result to the desired binary classification. A bidirectional wrapper will be included, which passes the input sequence both forward and backward through the LSTM network. Since Tagalog sentences have a flexible structure, often switching the order of the subject, verb and object, information before and after a set of words are equally important in understanding it. To further improve its performance, the LSTM layer will include dropout, which is the random removal of units from the neural network during training to reduce overfitting to the training set. The sigmoid activation function, which returns a value associated with the probability of the news article being fake, will be used for the final layer.

Fitting and Evaluation

The optimizer and loss function will be common to all created models. Everytime our model runs, we can utilize the data on the differences between the actual and predicted labels to tweak the weights of the connections between our neurons. We used the Adam optimizer to tweak the weights of our features. Next, we need a loss function that minimizes a particular value, in our case, the difference between the predicted and actual labels. We used the Categorical Cross Entropy Loss Function from Keras, which is the most popular loss function for categorical models. Since we're doing a binary classification experiment, classifying a news article as being fake news or not, we use binary cross entropy, which determines the cross entropy loss between our actual labels and the predicted probability in the output nodes. In order to evaluate our neural network model and compare its performance with other models, we need to obtain metrics. In this case, we calculated the accuracy of our model in predicting the label of either being fake news or not.

Lastly, we can fit our model and determine its accuracy in predicting the labels. Tidymodels enable us to split our data and create validation sets from our training data. We

specify the number of epochs or times our model trains on our data as well as the batch size to limit the number of data instances per epoch. After fitting our model, we can compare the accuracy of our dense neural network over time and determine the epoch at which the model is most accurate. Accuracy can go down over time because of overfitting.

Device Specifications

The models are run on the researcher's laptop computer device with the following specifications: CPU Intel Core i5-1135G7 (4 Cores, 8 Threads, 2.42 GHz), GPU Intel Iris Xe, 8GB RAM. The software used to run the models is RStudio.

IV. Results

The main metrics used to evaluate the three models are accuracy, precision, loss (binary cross entropy), sensitivity, specificity, and kappa. For each of the three models, we tested its performance on training data, validation data, and finally the best model was tested on the testing data.

For our data, positive refers to fake news while negative refers to real news. Accuracy is a metric which measures how many real news and fake news articles were correctly classified by the model as being real or fake. Precision measured the model’s ability to correctly predict fake news everytime it labeled an article as fake. As mentioned above, each of our models uses a loss function, binary cross entropy, common for binary classification tasks, that gets the difference between the actual labels and the predicted probabilities. Our model trains to gradually decrease the amount of loss in its resamples. Sensitivity and specificity were used when we created ROC curves as part of our model evaluation. The closer the area under the curve is to 1, the better the model. Sensitivity is the true positive rate while sensitivity is the true negative rate.

Model accuracy over epochs

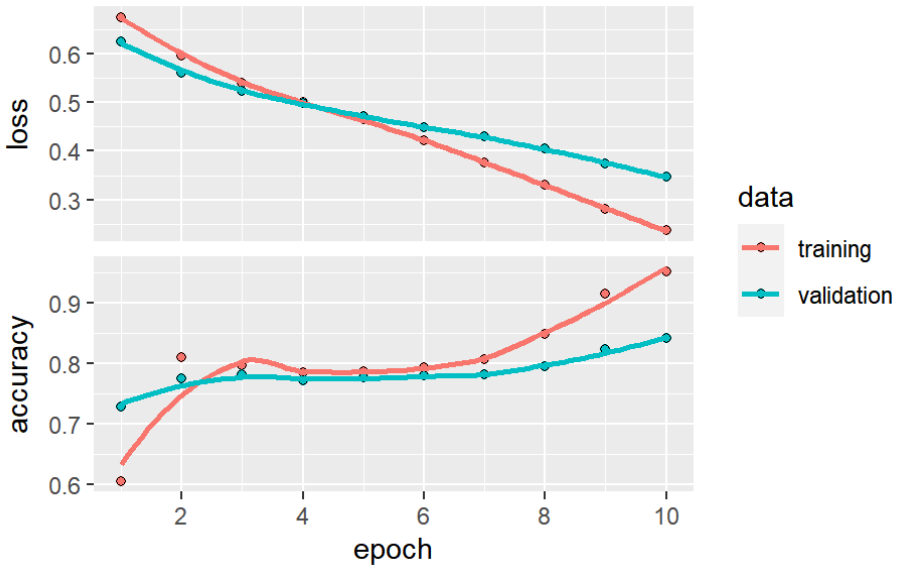

Figure 1. The graph of our DNN's accuracy and loss metrics based on the training and validation sets

The figure above shows that our dense neural network model wasn't really hindered by overfitting to training data. As the model kept training on different resamples, its accuracy on the validation data continually increased. Moreover, the top graph shows that the loss function decreased over the epochs.

```
Final epoch (plot to see history):
        loss: 0.2366
    accuracy: 0.9517
    val_loss: 0.3473
val_accuracy: 0.8405
```

Figure 2. The performance metrics of the 10th epoch of the DNN model

Overall, the initial 2 layer dense neural network had good metrics. The dense neural network had a very high accuracy of 95.17%, which means that it was very good at predicting the training data labels. The model's accuracy on the validation was still respectably high at 84.05%. It is common that the model's accuracy on the training data is significantly higher than its accuracy on the validation data.

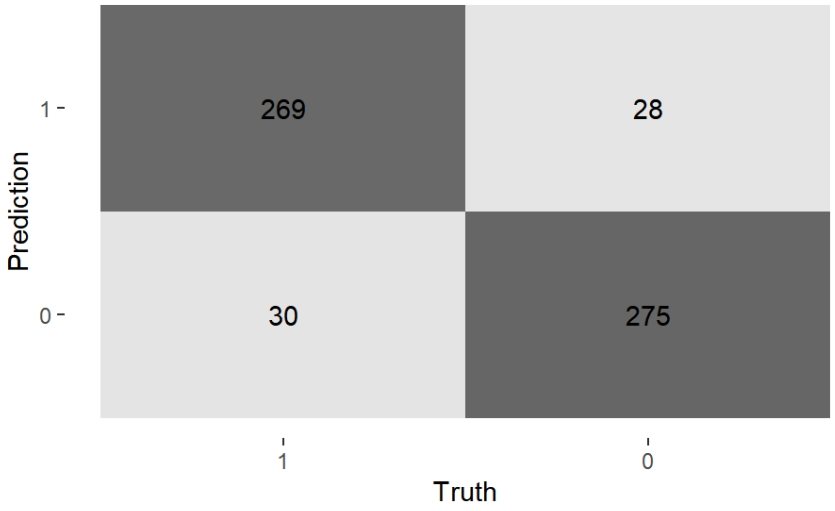

Figure 3. The Confusion Matrix for the DNN model on the validation data

The confusion matrix highlights that the boxes in the main diagonal: the true positives and true negatives were correctly predicted by the DNN model for the most part. The number of false positives, 28, and the number of false negatives, 30, was strikingly similar. This shows that the model was adequate at minimizing both false positives and false negatives.

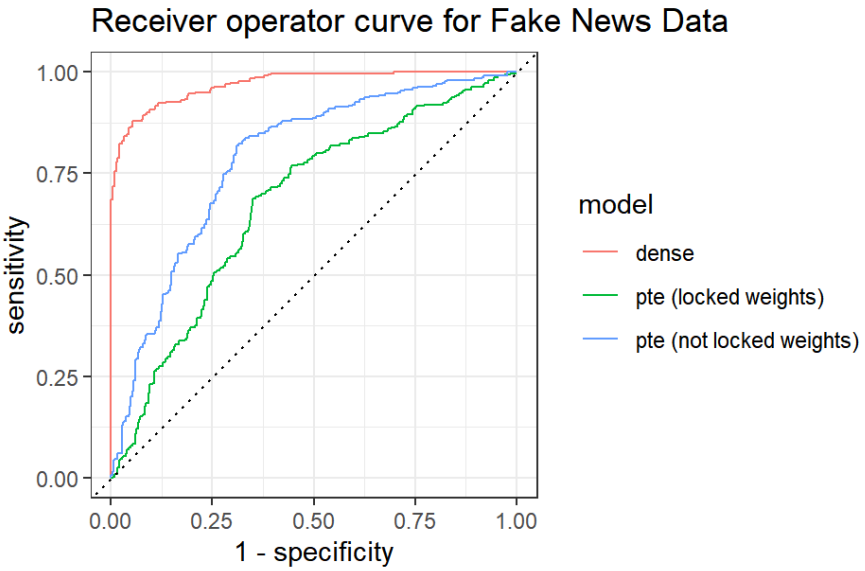

Figure 4. Comparison of the ROC curves on the validation data

model <chr>	.metric <chr>	.estimator <chr>	.estimate <dbl>
dense	accuracy	binary	0.9036545
pte (locked weights)	accuracy	binary	0.6362126
pte (not locked weights)	accuracy	binary	0.7392027
dense	kap	binary	0.8072920
pte (locked weights)	kap	binary	0.2721521
pte (not locked weights)	kap	binary	0.4786930

Figure 5. Comparison of the models based on accuracy and kappa

As we can see from figures 4 and 5, the 2 layer dense neural network without trained word embeddings showed the best performance. It had the highest accuracy and kappa by a significant amount. The ROC curves provide a visual representation of our model's effectiveness at distinguishing true positives from false positives. Likewise, the area under the curve of the base dense neural network was the greatest, signifying that it was the most effective and best trained model.

Final Evaluation

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.9002494
kap	binary	0.8002640
mn_log_loss	binary	0.2376600
roc_auc	binary	0.9652439

Figure 6. Final performance on the testing data

Finally, we evaluated the best model, the dense neural network with self-trained word embeddings, on the testing data. The metrics on the testing data are noticeably better compared to the metrics on the validation data. This discrepancy (90% accuracy on testing data and 84% accuracy on the validation data) can be attributed to random differences on the data. Overall, the dense neural network was very good at classifying the news articles as it showed good performance while taking less than 5 minutes for all the functions to run completely.

LSTM: Long Short-term Memory Neural Network

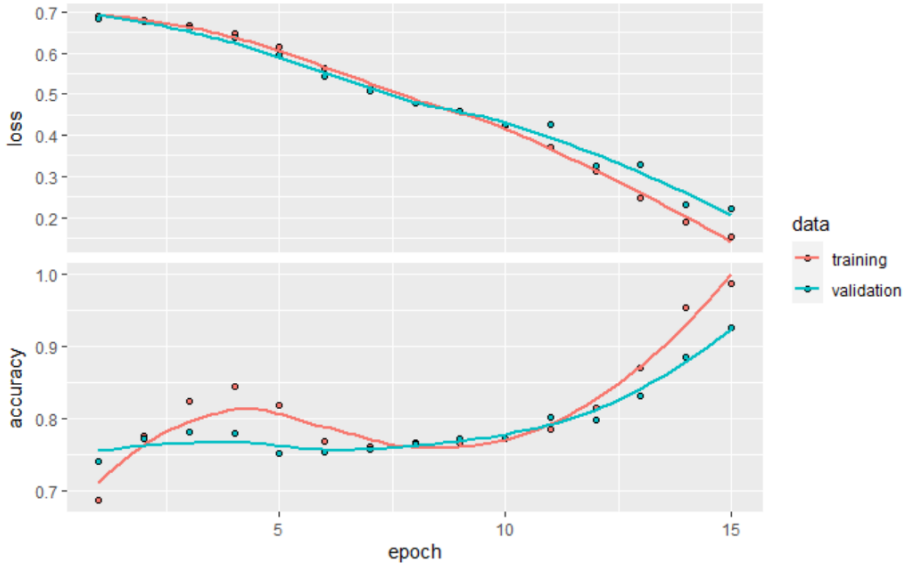

Figure 7. Accuracy and loss metrics for LSTM on training and validation set over 15 training epochs.

Initially, the LSTM showed a low accuracy for 80% with 10 training epochs. However, it was noticed that the accuracy was still increasing per epoch and the loss value between the training and validation set did not substantially differ by the 10th epoch. Thus, the decision was made to change the number of epochs to 15, which led to a model with a final accuracy of 92% on the validation set.

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.9127182
kap	binary	0.8254310
mn_log_loss	binary	0.2788496
roc_auc	binary	0.9593595

4 rows

Figure 8. Metrics for LSTM model on testing set.

The performance on the testing set was not significantly different from the performance during training, which shows that overfitting was minimal. As compared to the

DNN model, the performance of the LSTM model shows the ability of the model to learn patterns in sequential data as well as context of a word from before and after that word through bidirectionality. Dropout was also successfully introduced to reduce overfitting of the data, which can be seen in the loss metric of training and validation set which did not significantly differ through the course of 15 epochs. However, we note that the LSTM model training time took around 200 seconds which is much longer than the other created models.

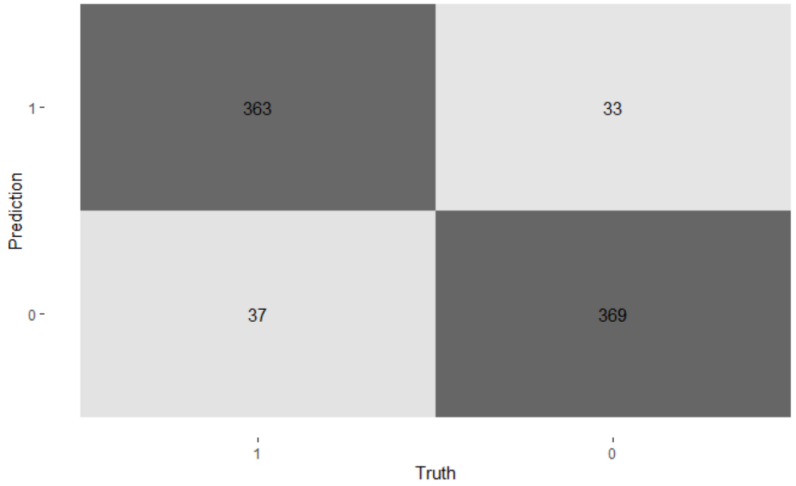

Figure 9. Confusion matrix for LSTM model on testing data.

Based on the confusion matrix, the LSTM model was able to correctly classify 363 fake news articles as fake and correctly classify 369 real news articles as real. The number of incorrect classifications, 33 false positives and 37 false negatives indicate that the LSTM model was biased in classifying fake or real news.

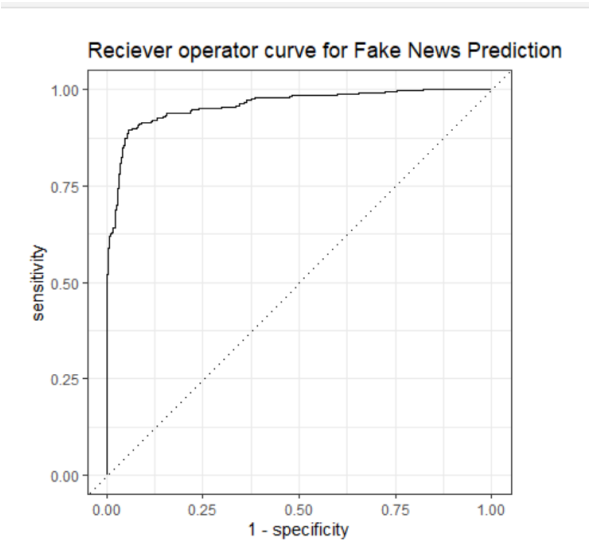

Figure 10. ROC curve for LSTM model on testing data.

The steep rise in the sensitivity of the ROC curve without introducing too many false positives shows that the LSTM model was able to correctly classify most of the fake news samples as fake without labeling that many real news as incorrectly fake. The area under the ROC curve was almost 0.96, which indicates that the model was incredibly accurate.

CNN: Convolutional Neural Network

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.9301746
kap	binary	0.8603448
mn_log_loss	binary	0.2026349
roc_auc	binary	0.9770771

4 rows

Figure 11. The performance metrics of the CNN model on the testing set

The performance metrics of the CNN showed the best results compared to DNN and LSTM. Due to its ability to learn local, spatial structure within data, it obtained an accuracy of 93% and an ROC of 97.7%.

The loss accuracy chart shows the number of epochs is directly related to the accuracy, while indirectly to the loss. A closer look at the loss reveals that at lesser epochs, the training data initially has more loss, but starting from 6 epochs, the training data starts to have significantly less loss. As for the accuracy portion of the chart, the accuracy of both the training and validation sets increases significantly until it starts to plateau starting from the 18th epoch. It is also notable that the training and validation are also similar, which implies that there was no overfitting.

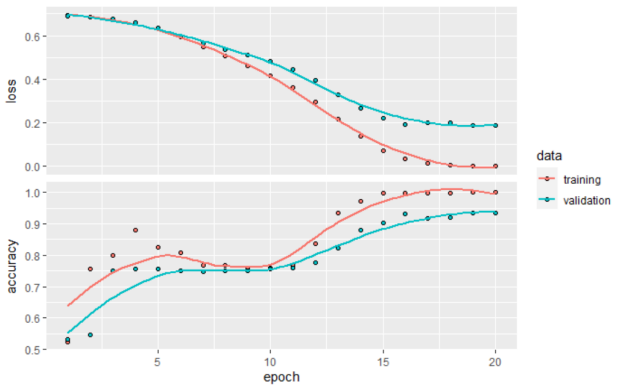

Figure 12. Accuracy and loss metrics for CNN on training and validation set over 20 training epochs

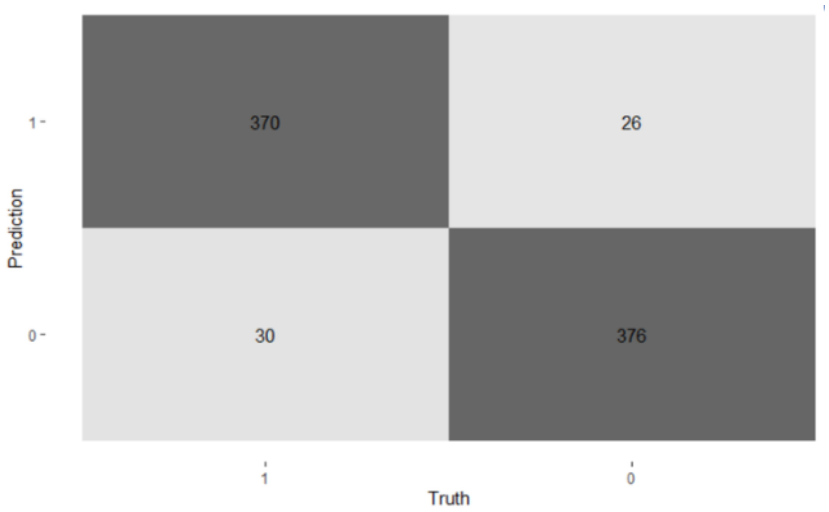

Figure 13. Confusion matrix for CNN model on testing data

The confusion matrix shows that the model was able to correctly classify fake and real news. 370 were classified as true positives, while 376 were classified as true negatives. The darker shades indicate that the majority of the data were classified in that area.

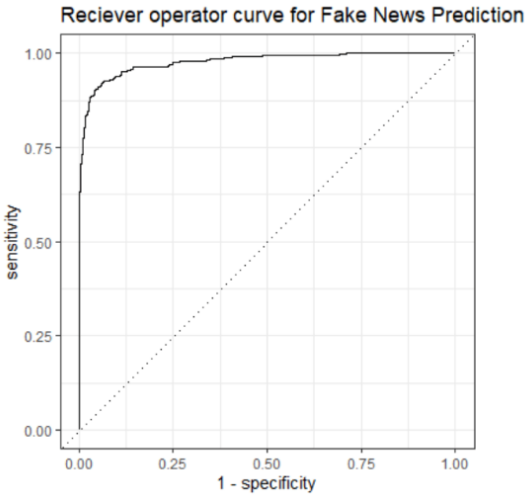

Figure 14. ROC curve of the CNN final model on the testing data

The area under the ROC curve of the CNN model was 0.977, which was higher than the area under the curve of the LSTM and the DNN models. This shows that the CNN model was the

best at increasing the true positive rate or labeling all the positive samples without making false positive errors.

Comparison of Models

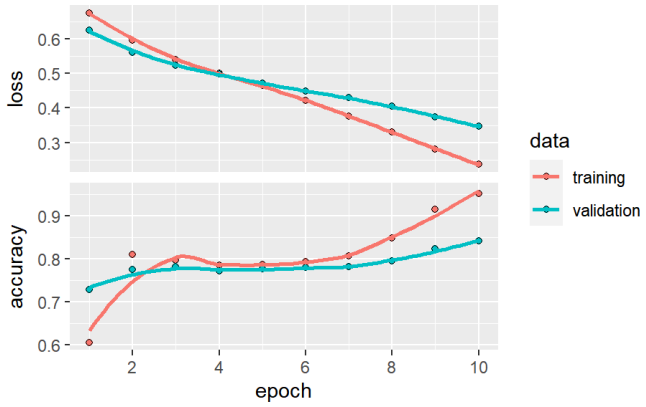

Figure 15a. Accuracy and loss metrics for DNN on training and validation set over 10 training epochs.

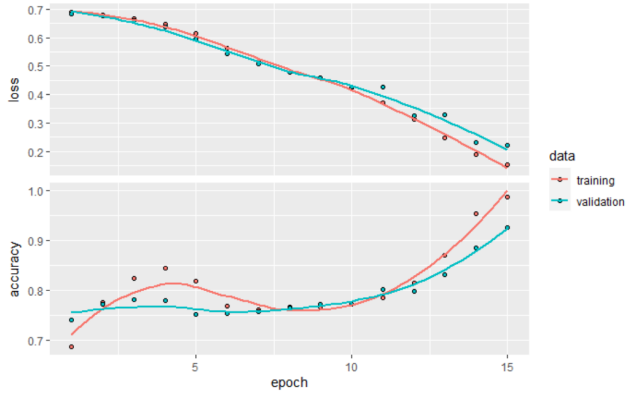

Figure 15b. Accuracy and loss metrics for LSTM on training and validation set over 15 training epochs.

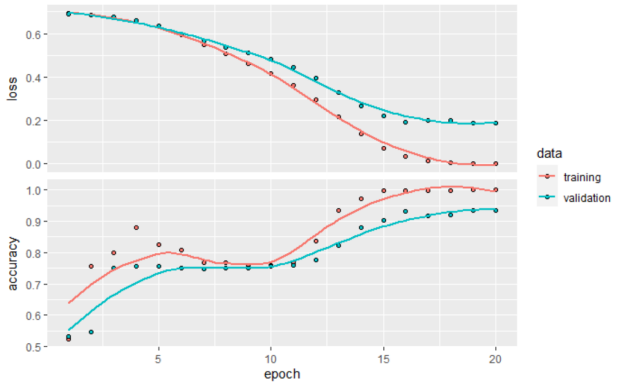

Figure 15c. Accuracy and loss metrics for CNN on training and validation set over 20 training epochs.

Over the training epochs, we observed the loss steadily decrease and the accuracy steadily increase for both the training and validation set. There is a period of slow improvement in accuracy before the final improvement to >90%. The accuracy settled down

after that, with further epochs no longer leading to an improvement past the epoch when the accuracy on the training set reached ~99%. This indicates, along with the loss metric, that overfitting has occurred on the model.

	Test accuracy	Kappa value	Loss value	Training time
DNN	90.02494%	0.8002640	0.2376600	2 seconds
LSTM	91.27182%	0.8254310	0.2788496	180 seconds
CNN	93.01746%	0.8603448	0.2026349	20 seconds

Figure 16. Accuracy metrics and Training time for all three neural network models.

Model	Val. Accuracy	Loss	Val. Loss	Pretraining Time	Finetuning Time
Siamese Networks	77.42%	0.5601	0.5329	N/A	4m per epoch
BERT	87.47%	0.4655	0.4419	66 hours	2m per epoch
GPT-2	90.99%	0.2172	0.1826	78 hours	4m per epoch
ULMFiT	91.59%	0.3750	0.1972	11 hours	2m per epoch
ULMFiT (no LM Finetuning)	78.11%	0.5512	0.5409	11 hours	2m per epoch
BERT + Multitasking	91.20%	0.3155	0.3023	66 hours	4m per epoch
GPT-2 + Multitasking	96.28%	0.2609	0.2197	78 hours	5m per epoch

Figure 17. Accuracy metrics and Training time for
neural networks trained by Cruz et al. (2020)

Interestingly, our three neural networks managed to achieve similar accuracy to the neural networks in our reference paper by Cruz et al. (2020), despite their neural networks taking much longer to train (180 seconds versus 11 hours). We note that this may be explained by a difference in device specifications; the reference paper used a single Google Tensor Processing Unit sourced from cloud computing, while we used a full quad-core Intel Core processor with integrated graphics.

V. Conclusion

While all three neural networks (DNN, LSTM, CNN) showed great overall performance, the convolutional neural network was the best model in terms of accuracy, loss, and kappa in predicting and classifying the fake news Filipino language dataset. When compared to the reference study of Cruz et al., our neural network was able to achieve similar or even better performance in terms of accuracy and loss while being vastly more efficient and faster to

train. The use of byte pair encoding on the model's preprocessing recipe improved its accuracy and performance.

The convolutional neural network had a relatively fast processing and training time compared to the long short term memory model and it was just barely slower than the basic dense neural network model. The presence of the kernel and multiple convolutional layers may have been factors which strengthened the performance of the CNN model in classifying the dataset. Convolutional networks focus on determining and recognizing patterns in the spatial position of data. CNNs are also most suited for classification tasks like sentiment analysis and spam detection (Ghelani, 2019). These examples are similar to this research's goal of classifying news articles. Moreover, the convolutional neural network's training over 20 epochs helped it better classify the news sample data without being subject to the negatives of overfitting.

VI. Recommendations

The researchers have a couple of recommendations to expand the scope, significance, and accuracy of our classification models. First, future studies can include more features to the recipe pre-processing such as by adding Filipino stopwords data and by tuning the hyperparameters (sequence length, vocabulary size, and kernel size for CNN). Moreover, the dataset of fake news articles could be expanded to include opinion pieces, satirical articles, and even showbiz and entertainment. Other forms of information like tweets, social media posts, tabloids, and memes could be explored and classified by more advanced classification models.

The models could be tested on a new dataset of articles to ensure that the models are adaptable and suited for general use. Moreover, the study can be expanded to include other Filipino dialects and language if relevant datasets can be sourced. It would also help if more relevant and timely news articles could be added to the existing dataset, considering that fake news has become even more prevalent than ever in the past few years.

Bibliography

Cruz, J. C. B., Tan, J. A., and Cheng, C. (2020, July 1). Localization of Fake News Detection via Multitask Transfer Learning. *Proceeding of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2596-2604. https://aclanthology.org/2020.lrec-1.316.pdf

Dobilas, S. (2022, March 5). *LSTM Recurrent Neural Networks — How to Teach a Network to Remember the Past*. Medium. https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e

Ghelani, S. (2019, June 2). *Text Classification—RNN's or CNN's?* Medium; Towards Data Science. https://towardsdatascience.com/text-classification-rnns-or-cnn-s-98c86a0dd361

Kumar, A. (2020, October 28). *Keras - Categorical Cross Entropy Loss Function*. Data Analytics. https://vitalflux.com/keras-categorical-cross-entropy-loss-function/

Lalu, G. P. (2022, October 11). *'fake news' a problem in ph? 9 in 10 Filipinos agree, says pulse Asia*. INQUIRER.net. Retrieved November 27, 2022, from https://newsinfo.inquirer.net/1678248/fake-news-a-problem-in-ph-9-in-10-filipinos-agree-says-pulse-asia?fbclid=IwAR0eh-qMvTIyjzQ_9o-FUaKqWF531aTKfhgynfqiat6a0hh7_qpvfMm6ADs

Siar, S. (2021, August). *Fake news, its dangers, and how we can fight it*. Retrieved November 27, 2022, from https://pidswebs.pids.gov.ph/CDN/PUBLICATIONS/pidspn2106.pdf

Silge, E. H. and J. (n.d.). Chapter 10 Convolutional neural networks | Supervised Machine Learning for Text Analysis in R. In *smltar.com*. Retrieved November 27, 2022, from https://smltar.com/dlcnn.html

De Paor, S., & Heravi, B. (2020). Information literacy and fake news: How the field of librarianship can help combat the epidemic of fake news. *The Journal of Academic Librarianship, 46*(5), 102218. https://doi.org/10.1016/j.acalib.2020.102218

Hvitfeldt, E., & Silge, J. (2022). Supervised machine learning for text analysis in R. CRC Press.