Assignment 3: Deep Learning with NLP

Gavyn Gallagher

Northwestern University

MSDS 458: Artificial Intelligence and Deep Learning

Syamala Srinivasan

November 6, 2022

Abstract

Company X is seeking to reorganize all of its documents and processes. This will allow team members to search and find information faster and increase productivity. A model is asked to be created to classify Company Xs' text data. To create the model, the data set AG News was used. AG news holds thousands of new articles of various types. In this assignment 11, neural network models that went through text vectorization were used. There were different types of neural network models: RNNs, LSTMs, and CNN. LSTM models proved to score the best. The highest model achieved 86.02% accuracy. A recommendation is that Company X continue working on improving the LSTM model with new parameters to achieve higher accuracy. Once a high enough accuracy is achieved then the model can be implemented to reorganize the documents.

Introduction and Problem Statement

Our Company X has a new initiative to reorganize all the company's documentation into categories. Ideally, finance documents would fall under finance and marketing under marketing. Once all of the documents are in the correct location it will be easier to find important information. Company X asks us to design and implement a model that can achieve this process.

To test our model, the data set called AG news subset will be used. The data set is over 1 million news articles from over 2000 sources. All of the data is text data. Each of the articles belongs to one of four categories: world, sports, business, sci/tech. Representation of each category is split evenly. Theoretically, if we can develop a model that classifies the AG news subset text data then that model should work on Company Xs' documents. Once a model reaches a threshold of accuracy it will be implemented.

In this assignment, four experiments were conducted resulting in eleven distinct models being created. Experiment A, was an exploratory data analysis of the AG data with the goal to learn more about data and classes. There was also tweaking of the vocabulary size, removal of common words, and utilization of a fixed number out sequence length. Experiment B begins the creation of models with five RNN models being built. Each of the five with slight variations such as unidirectional/bidirectional, regularization, and fixed number out sequence length. Similar Experiment C uses those same variations but on LSTM models. Lastly, in experiment D, a 1 dimensional CNN was created.

In the data preparation, training and testing data will be split with 120,000 samples in training and 7,600 samples in testing. All models will be evaluated on accuracy and loss scores. The best model will be recognized and recommendations for what Company X should do next will be stated.

Literature Review

Sequential Models for Text Classification Using Recurrent Neural Networks

This article by Winda Kurnia Sari and her team focuses on using RNN and LSTM on the AG news dataset(SARI, 2020). The goal was to use text vectorization and neural networks to achieve high accuracy in classifying the 4 types of articles.

The article goes into depth about what RNN and LSTM are. Also discussed is how text vectorization and word embedding plays an important role. Tokenization, removal of punctuation, and one hot encoding were all used in the data preprocessing. Optimizers used in their include Adam and RMSProp.Results of their models show accuracy running in the mid-nineties.

This is an important article as it demonstrates how to use RNN and LSTM well on the AG dataset. The preprocessing section was helpful and many of the methods there will be used in this assignment. We hope to achieve similar successful results as this article.

Generative and Discrimatintive Text Classification with Recurrent Neural Networks

Dan Yogatama and others wrote this article to discuss how generative and discriminative models are different (Yogatama, 2017). AG news dataset was one of a few datasets used in this article. The main architecture used was LSTM with generative and discriminative LSTMs being created.

The article details the pros and cons of generative and discriminative models. Where discriminative models over rely of the newly made iteration and suffer from not learning from previous iterations well. While the generative model's goal is to generalize the data well. The authors concluded the generative model did well on smaller datasets but struggled with the complexity of larger datasets.

This article demonstrates the vastly different types of LSTMS that can be created. In this assignment we plan to keep the architecture relative the same for each model, only testing one parameter at a time. We hope to build superb architecture for the model and will have to consider what the architecture should entail.

Method(s)

Data Preparation was performed on the AG data set. First, the data was standardized by converting the text data to lowercase, and then removing special characters such as ‘!@#$%’ and more. Next common words called stopwords such as (a, the, and, I) were removed. These words add little to the analysis as they are not nouns or verbs and with how common they are they can

skew the results. Text vectorization was then used to split the text data into tokens or the lowest level of granularity. The train and test data will be based on text vectorization.

In experiment A, an exploratory data analysis (EDA) was performed with three parts. In the EDA it was discovered that there is four classes of data: world, sports, business, sci/tech. With each class having an equal 31,900 samples. Vectorizations were done in several ways. In part a of experiment A the vocabulary size was set at 1,000, 5,000, and 10,000. This was to see the top words in each part as well as to see the percentage of non-vocabulary words in a document. In part b, common stop words were removed to see how that would influence the top words and the proportion of non-vocabulary words. Part C, used a fixed number of output sequence length of 100 which will be used to pad the text up to this length.

In experiment B, Recurrent Neural Network(RNN) models were built. RNN work by each component having the same weight and using internal memory where all the inputs are related to each other (Mittal, 2019). The 5 models in this experiment were differentiated by several parameters: bidirectional/unidirectional, regularization, and output sequence length being the default or fixed integer. Information can either flow unidirectionally or birdectionally. In birdection models the information goes forward and in reverse (Zvornicanin, 2022). Regualzation was also used in six of the ten models, is it used for simplifying the model and combatting overfitting (Haque et al., 2022). According to the Keras documentation for text vecotrization, output sequence length is a parameter that can be set to a fixed integrer to change the size of the output. Below lists out how each RNN model is set up differently.

- RNN_model1: Bidirectional, no regularization, output sequence length = default
- RNN_model2: Unidirectional, no regularization, output sequence length = default

- RNN_model3: Bidirectional, regularization, output sequence length = default
- RNN_model4: Unidirectional, regularization, output sequence length = default
- RNN_model5: Bidirectional, regularization, output sequence length = 150

In experiment C, Long term short memory(LSTM) models were built. LSTM is a modified RNN that solves the gradient descent issue by using gates remember long term memory (Mittal, 2019). The gates are the input, forget, and output gates (Mittal, 2019). The 5 models in this experiment were differentiated by several parameters: bidirectional/unidirectional, regularization, and output sequence length being the default or fixed integer.

- LSTM_model1: Bidirectional, no regularization, output sequence length = default
- LSTM_model2: Unidirectional, no regularization, output sequence length = default
- LSTM_model3: Bidirectional, regularization, output sequence length = default
- LSTM_model4: Unidirectional, regularization, output sequence length = default
- LSTM_model5: Bidirectional, regularization, output sequence length = 150

In experiment D, a one-dimensional convolutional neural network(CNN) model was built. CNN (citation). Unlike previous experiments, there was only one model built. Parameters of the model included bidirectional, regularization, and a default output sequence length.

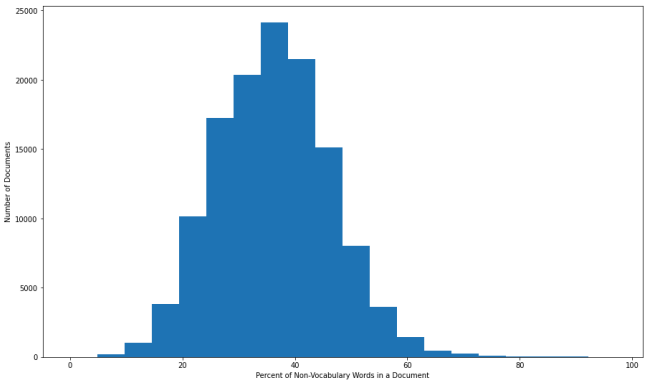
Results

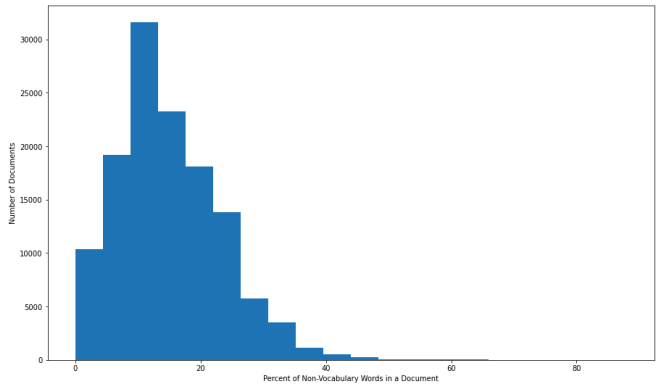
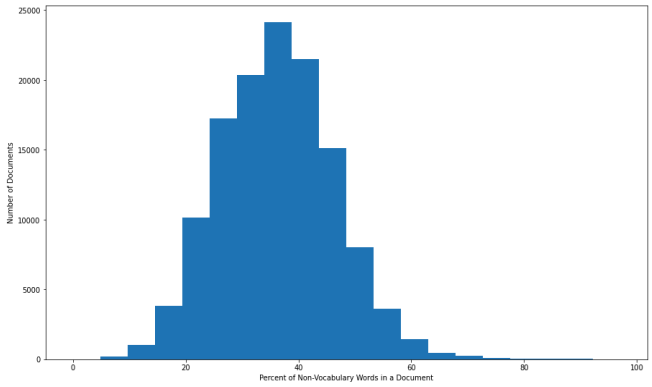
Experiment A

Figure 1
AG data set most common words

```
[('the', 183490),
 ('to', 100936),
 ('a', 100359),
 ('of', 94260),
 ('in', 80008),
 ('and', 69279),
 ('on', 49107),
 ('-', 40397),
 ('for', 39128),
 ('that', 28545),
 ('#39;s', 26499),
 ('The', 25280),
 ('with', 22895),
 ('its', 22301),
 ('as', 21947),
 ('is', 20789),
 ('at', 20752),
 ('has', 19331),
 ('by', 18146),
 ('said', 17482),
 ('from', 16513),
 ('it', 16001),
 ('an', 15942),
 ('his', 15045),
 ('will', 14122)]
```

Figure 2
Different vocabulary sizes percentage of Non-Vocabulary words in a document

Vocabulary Size	% of Non-Vocab Words in a Document By Number of Documents
1,000	

5,000	
10,000	

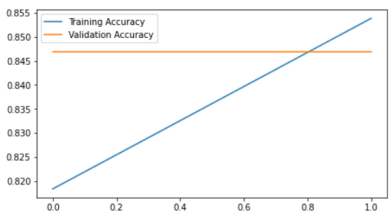
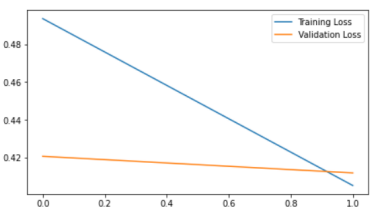
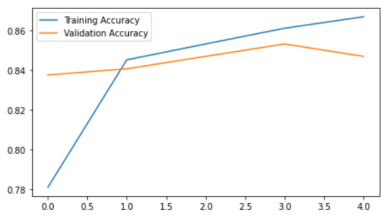
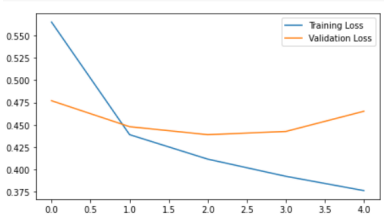
Experiment B

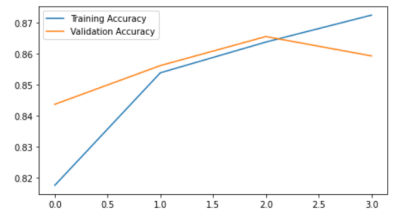
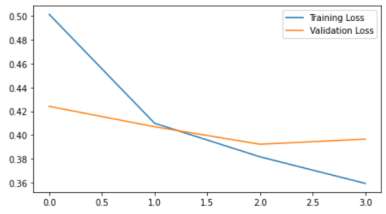
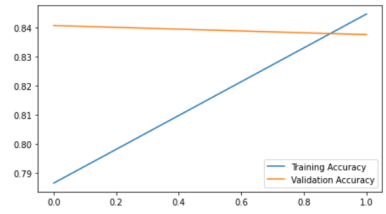
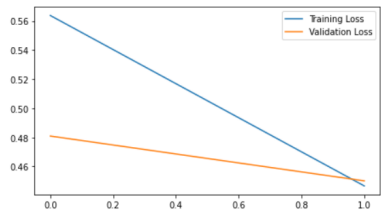
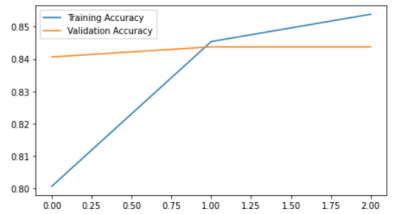
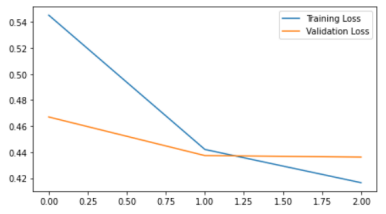
All models in this assignment will have accuracy and loss graphs. These graphs show the training and validation accuracy and loss. RNN model's accuracy and loss plots can be seen in figure 3. The number of iterations or epochs is displayed on the X-Axis. Even though all models were said to be 200 epochs, through early stopping some models have more iterations than others. Interestingly the models did not run many interactions with the most being RNN_model2 with 4 iterations. There is no distinction between models with bidirectional or regularization when it comes to the number of iterations ran for these models.

When it comes to scores, the models all performed well and similarity to each other. It is highly interesting how small of a range it was from the best model to the worst. This is likely due to all of the models having the same RNN architecture. Meaning the difference in scores between the models was the changed parameters. All of the models had higher training accuracy than validation accuracy. When it comes to loss, three of the models had lower training loss than validation loss.

- Highest Training Accuracy: RNN_model3
- Highest Validation Accuracy: RNN_model3
- Lowest Training Loss: RNN_model2
- Lowest Validation Loss: RNN_model3

Figure 3
RNN Models' Accuracy and Loss Plots

Model Name	Accuracy	Loss
RNN_Model1		
RNN_Model2		

RNN_Model3		
RNN_Model4		
RNN_Model5		

Moving on to see the accuracy and loss score for the test dataset, figure 4 shows how each RNN model performed. RNN_model1 had both the best scores for accuracy and loss. RNN_model2 performed the worst dsipate having the lowest training loss. In experimenting with directions, Bidirectional models had slightly higher accuracy and lower loss. Regularization did have a strong impact as RNN_modle3 did worse than RNN_model1. However, Regularization did improve the RNN_model4 over RNN_model3. Setting the output sequence length to a fixed number over the default did improve RNN_model5 over RNN_model3. The processing times for the models ranged. Setting the output sequence length to a fixed number doubled the amount of processing time for RNN_model5. Overall the RNN models were a success.

Figure 4
RNN Models' Scores

Model Name	Accuracy	Loss	Direction	Regularization	Output Seqeuence Length	Process Time
RNN_model1	0.8501	0.421	Bi	No	Fixed	3:53
RNN_model2	0.8337	0.4778	Uni	No	Fixed	5:10
RNN_model3	0.8479	0.4344	Bi	Yes	Fixed	6:49
RNN_model4	0.8411	0.4558	Uni	Yes	Fixed	2:18
RNN_model5	0.84355	0.4539	Bi	Yes	150	12:14

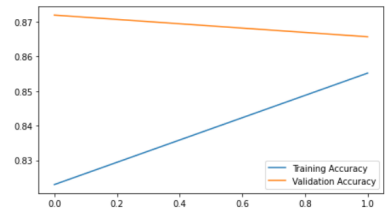
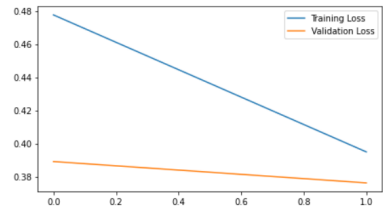
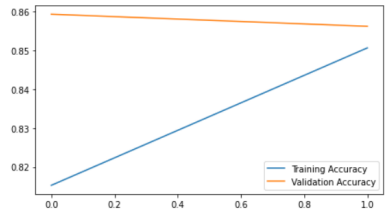
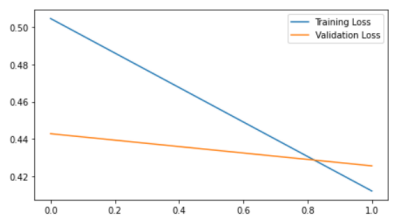
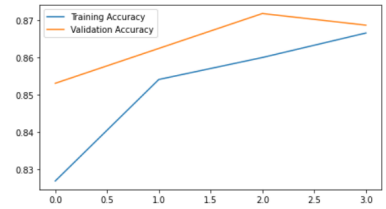
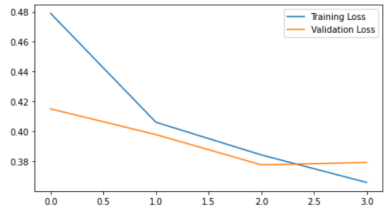
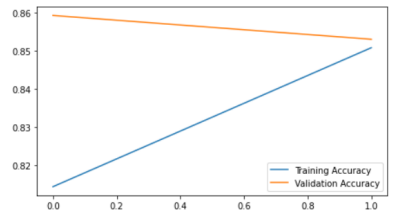
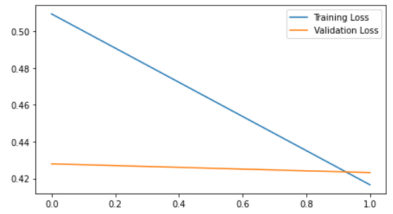
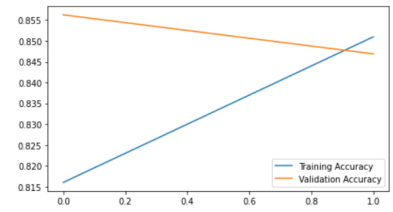
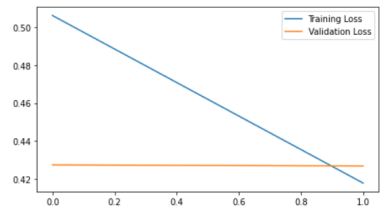
Experiment C

LSTM models were next conducted. Accuracy and loss plots of these models can be found in fgiure 5. The accuracy and loss scores performed similar yet slightly better to how the RNN models did. Only LSTM_model3 ran more than one iteration. This was advantageous and lead to LSTM_model3 having the best scores. Changed Paramters did not make much of an impact on the scores. Ideally the acrchietexute of the models would be changed to run more iterations.

- Highest Training Accuracy: LSTM_model3
- Highest Validation Accuracy: LSTM_model3
- Lowest Training Loss: LSTM_model3
- Lowest Validation Loss: LSTM_model1

Figure 5
LSTM Models' Accuracy and Loss Plots

Model Name	Accuracy	Loss

LSTM_Model1		
LSTM_Model2		
LSTM_Model3		
LSTM_Model4		
LSTM_Model5		

Turning to the accuracy and loss scores of the test data set. The LSTM models performed well and even slightly higher than the RNN models. Regularization improved the directional model LSTM_Model1 with LSTM_model3. Interesting how on unidirectional models in both RNN and LSTM did not improve with regularization. LSTM_model3 had a long process time compared to other models including its counterpart RNN_model3, this was for LSTM_mode3

ran more iterations. Using a fixed output sequence length did improve scores. Overall, LSTM models performed well and better than RNN.

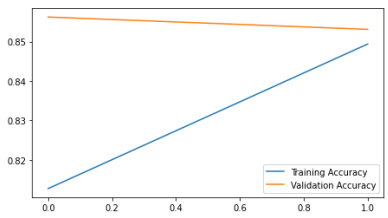
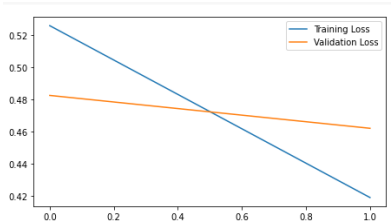
Figure 6
LSTM Models' Scores

Model Name	Accuracy	Loss	Direction	Regularization	Output Seqeuence Length	Process Time
LSTM_model1	0.8546	0.40462	Bi	No	Fixed	5:07
LSTM_model2	0.8526	0.4183	Uni	No	Fixed	4:01
LSTM_model3	0.8602	0.3918	Bi	Yes	Fixed	12:37
LSTM_model4	0.8483	0.4248	Uni	Yes	Fixed	3:27
LSTM_model5	0.8505	0.4267	Bi	Yes	150	10:56

Experiment D

In the final experiment a one-dimensional CNN model was created. This model has a convolutional layer and used similar architecture to the previous models. Bidirectional was not used nor was a fixed output sequence length. Accuracy and loss plots for the model can be seen in figure 7. The model only ran for one iteration. Generally, when this happens the scores are worse as the model can not learn from previous iterations.

Figure 7
CNN_model1 Accuracy and Loss Plots

Model	Accuracy	Loss
CNN_Model1		

To see the CNN_model1s accuracy and loss scores and information look to figure 8. With an accuracy score of .8493, the model ranks in the middle of the pack of all models. Loss perofmred worse, as this is the second lowest loss score of the eleven models. More CNN models should be made of various parameters to give CNN a fairer chance at achieving sufficient results.

Figure 8
CNN_model1s' Scores

Model Name	Accuracy	Loss	Direction	Regularization	Output Seqeuence Length	Process Time
CNN_model1	0.8493	0.4723	Uni	Yes	Default	00:30

Conclusion

In conclusion, neural network models were used to evaluate the text data of the AG news data set. If a model was proven to be a success then Company X would use the model for a reorganization of their own documents. Once Company Xs' documents are reorganized it will be easier for users to find the necessary documents they are looking for.

Five experiments were held and 11 models were ultimately created. 5 of which were for RNN and LSTM, while one model was for CNN. Unidirectional or bidirectional information flow proved to have little effect on model results. Models that ran more iterations tended to generate better accuracy and loss scores. Having a fixed output sequence length also added little help but drive increase the processing time significantly. Of the model types, LSTM models performed the best with 4 of the top five performing models coming from LSTM.

All models did moderate to well at classifying the 4 types of news articles. We recommend Company X to be happy with these results but keep testing before implementing as there is room for improvement. More iterations in models would be key. Explore more LSTM models as they proved to do the best. Change the regularization L2 rate. Since only one CNN model was made, we recommend making at least 4 more before writing it off. Now Company X has the groundwork to continue their NLP neural network to reorganize their documents.

Reference

Haque, R. U., Khan, R. H., Shihavuddin, A. S. M., Syeed, M. M. M., & Uddin, M. F. (2022). Lightweight and Parameter-Optimized Real-Time Food Calorie Estimation from Images Using CNN-Based Approach. *Applied Sciences*, *12*(19), 9733. https://doi.org/10.3390/app12199733

Mittal, Aditi. "Understanding RNN and LSTM." *Medium*, Medium, 26 Aug. 2021, https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e.

SARI, W. K., RINI, D. P., MALIK, R. F., & AZHAR, I. S. B. (2020, May). Sequential models for text classification using recurrent neural network. In *Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)* (pp. 333-340). Atlantis Press.

Yogatama, D., Dyer, C., Ling, W., & Blunsom, P. (2017). Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898*.

Zvornicanin, E. (2022, January 25). *Differences Between Bidirectional and Unidirectional LSTM | Baeldung on Computer Science*. Www.baeldung.com. https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm

Appendix

Figure 1
AG data set most common words

```
[('the', 183490),
 ('to', 100936),
 ('a', 100359),
 ('of', 94260),
 ('in', 80008),
 ('and', 69279),
 ('on', 49107),
 ('-', 40397),
 ('for', 39128),
 ('that', 28545),
 ('#39;s', 26499),
 ('The', 25280),
 ('with', 22895),
 ('its', 22301),
 ('as', 21947),
 ('is', 20789),
 ('at', 20752),
 ('has', 19331),
 ('by', 18146),
 ('said', 17482),
 ('from', 16513),
 ('it', 16001),
 ('an', 15942),
 ('his', 15045),
 ('will', 14122)]
```

Figure 2

Vocabulary sizes:1,000.Percentage of Non-Vocabulary words in a document

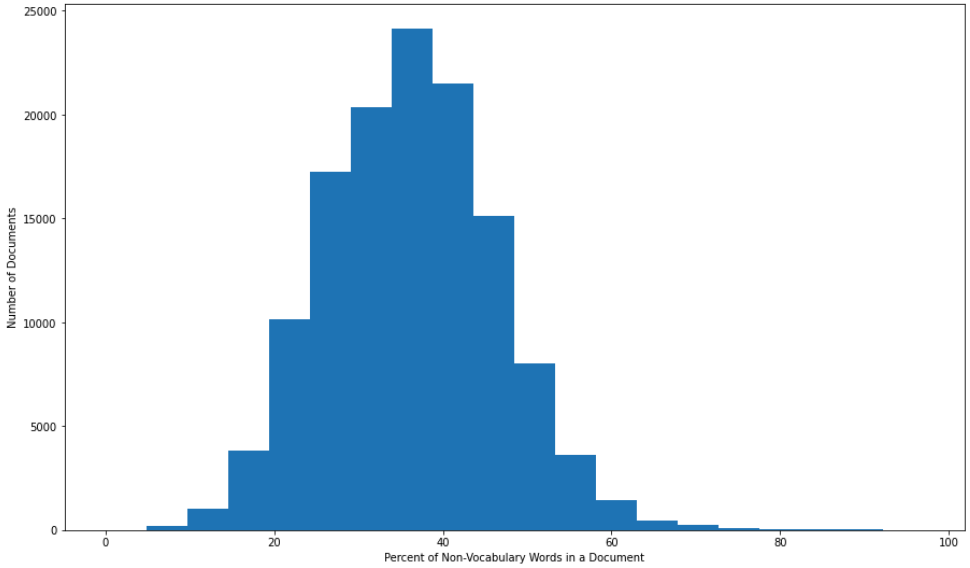


Figure 3

Vocabulary sizes:5,000.Percentage of Non-Vocabulary words in a document

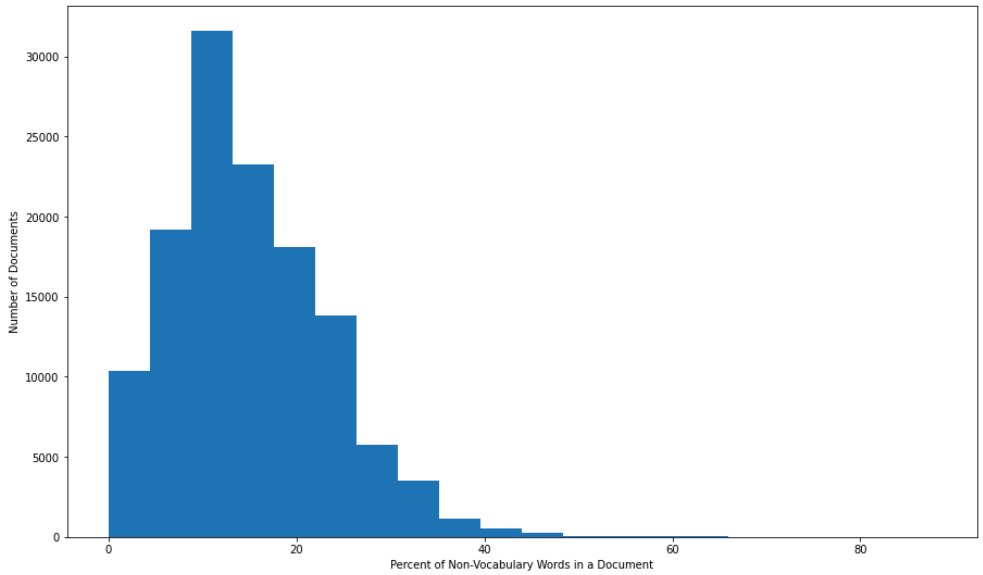


Figure 4

Vocabulary sizes:10,000.Percentage of Non-Vocabulary words in a document

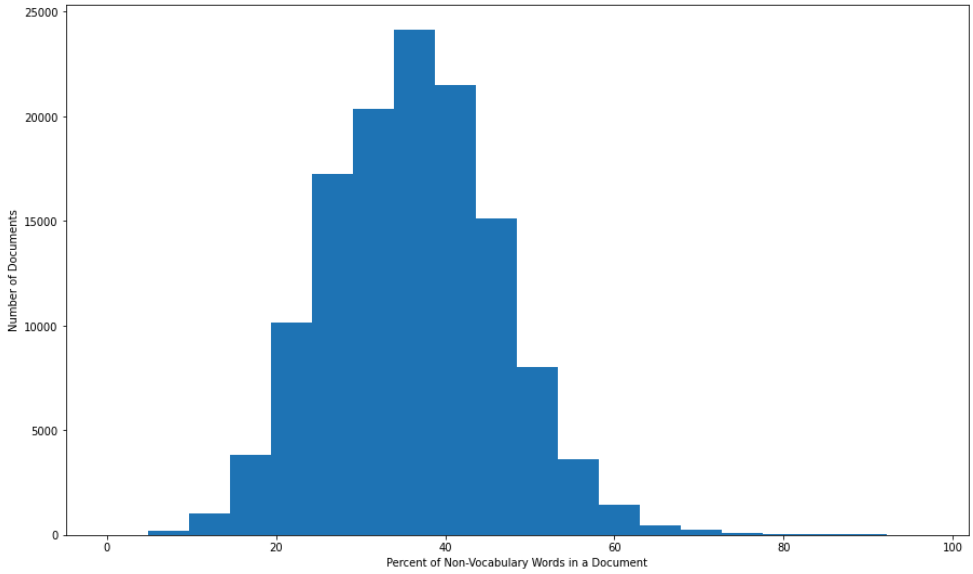


Figure 5
RNN_Model1 Accuracy plot

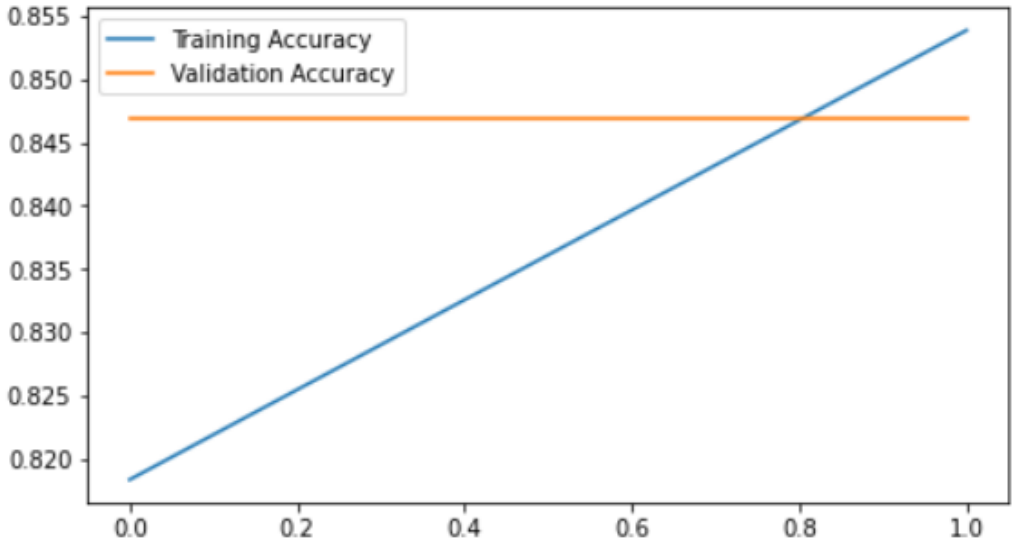


Figure 6
RNN_Model1 Loss plot

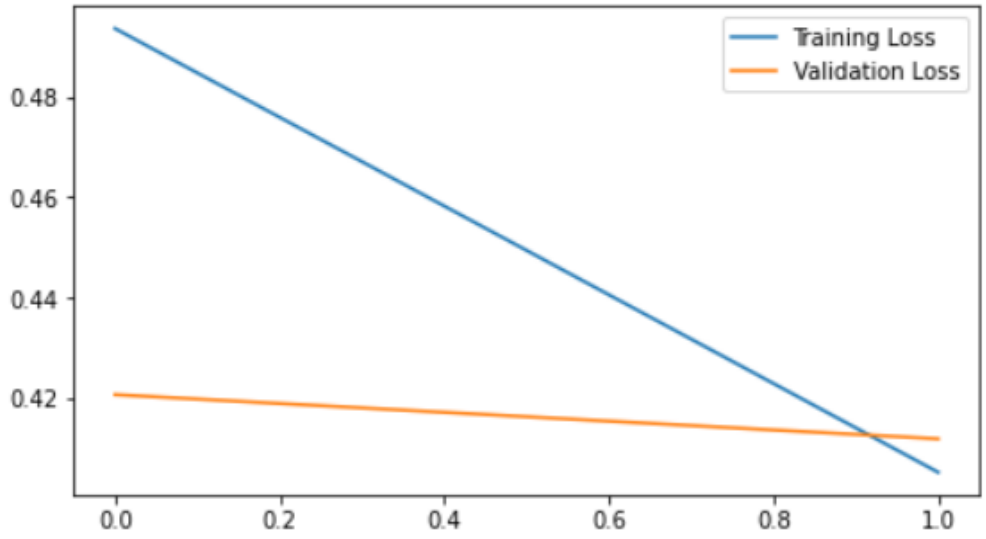


Figure 7
RNN_Model2 Accuracy plot

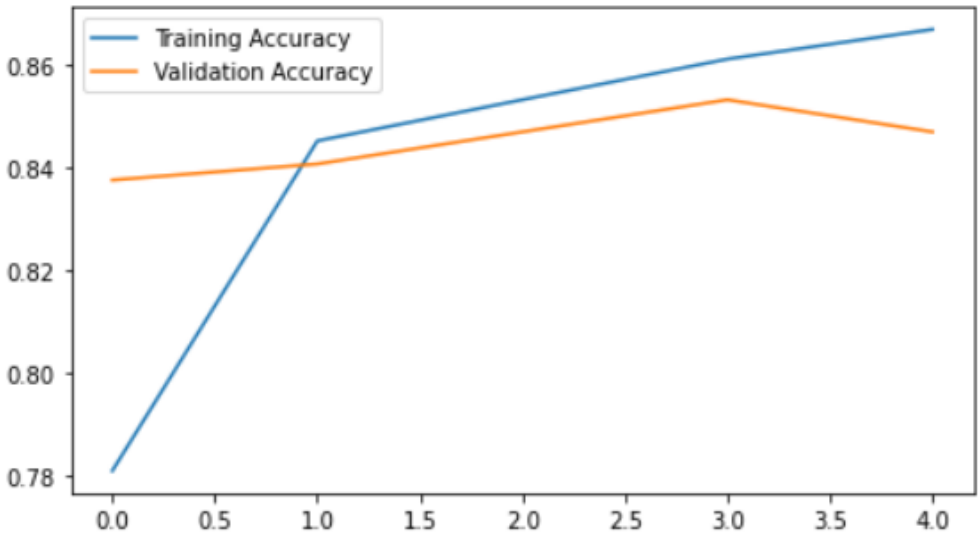


Figure 6
RNN_Model2 Loss plot

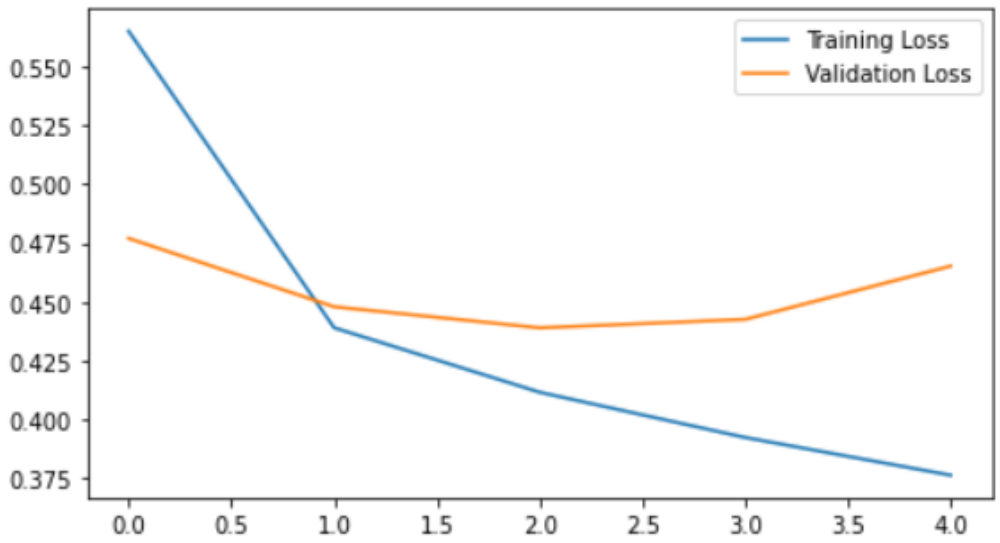


Figure 7
RNN_Model3 Accuracy plot

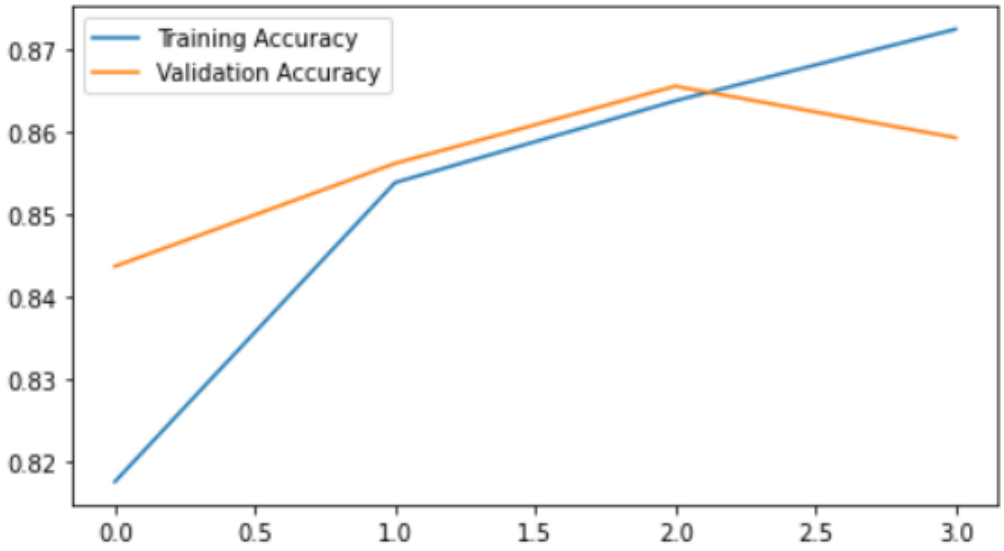


Figure 8
RNN_Model3 Loss plot

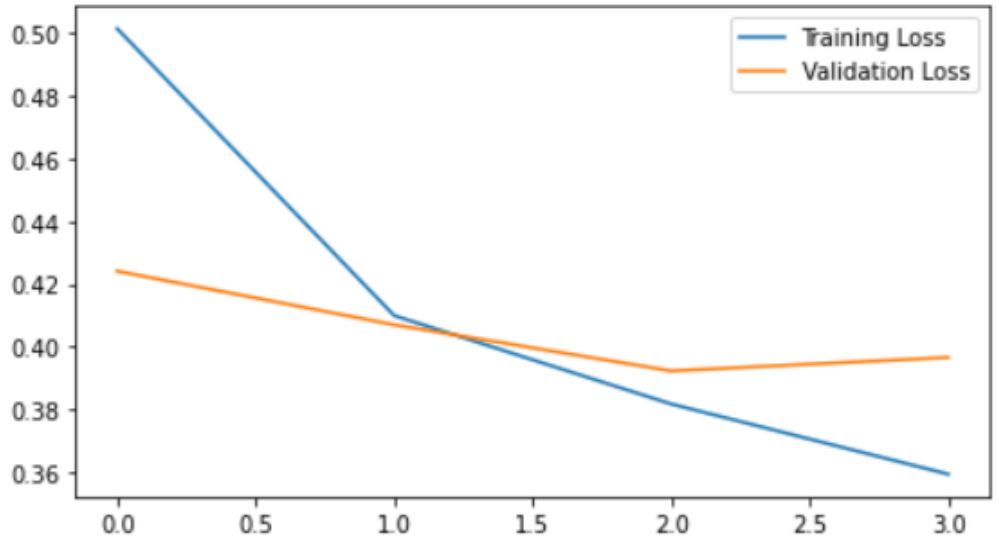


Figure 9
RNN_Model4 Accuracy plot

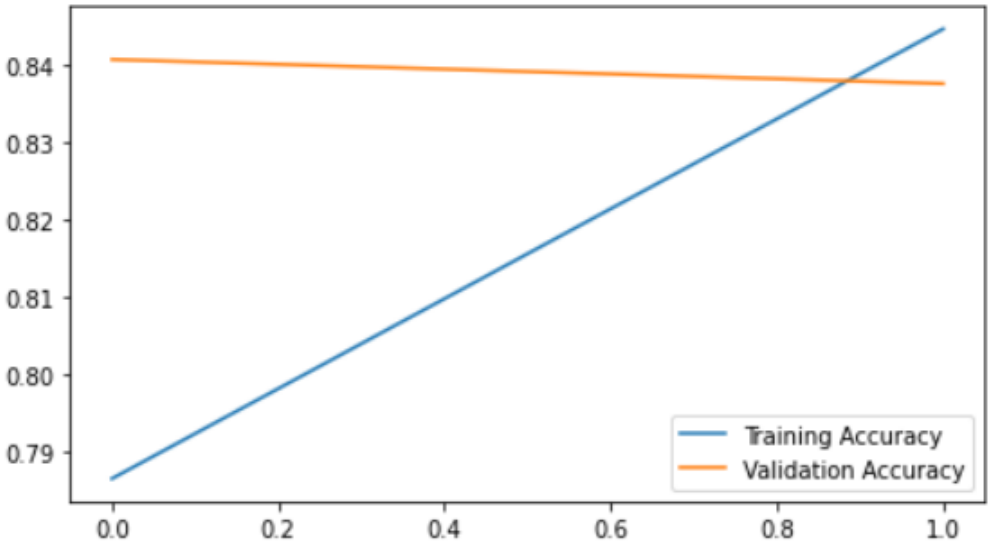


Figure 10
RNN_Model4 Loss plot

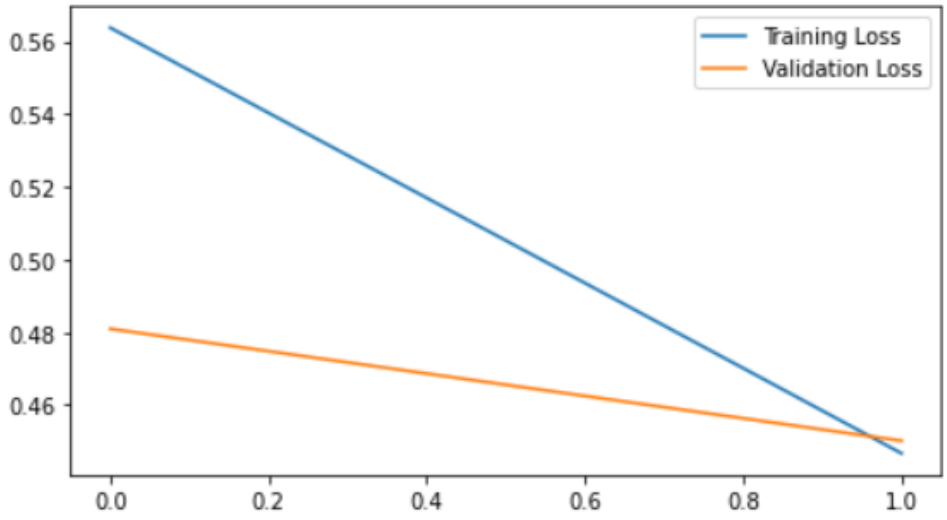


Figure 11
RNN_Model5 Accuracy plot

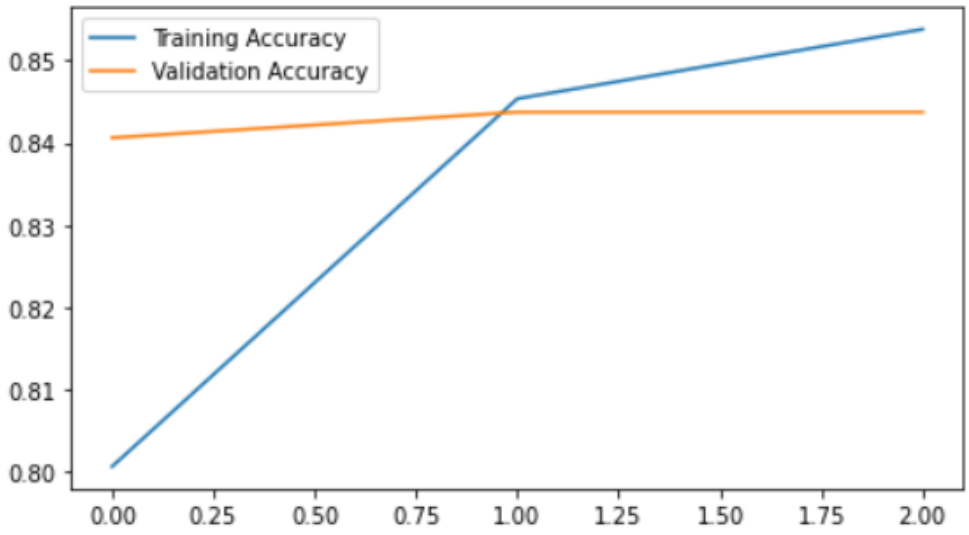


Figure 12
RNN_Model5 Loss plot

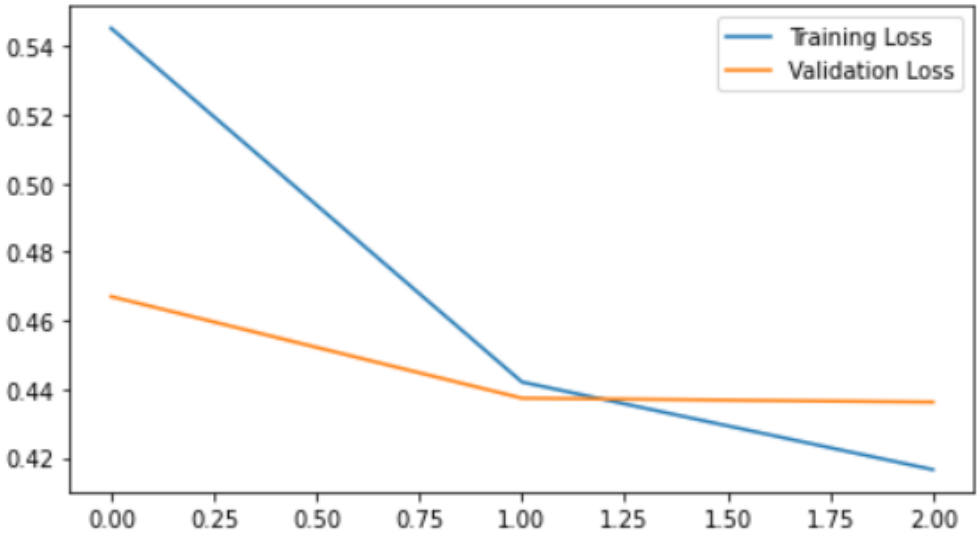


Figure 13
RNN Models' Scores

Model Name	Accuracy	Loss	Direction	Regularization	Output Seqeuence Length	Process Time
RNN_model1	0.8501	0.421	Bi	No	Fixed	3:53
RNN_model2	0.8337	0.4778	Uni	No	Fixed	5:10
RNN_model3	0.8479	0.4344	Bi	Yes	Fixed	6:49
RNN_model4	0.8411	0.4558	Uni	Yes	Fixed	2:18
RNN_model5	0.84355	0.4539	Bi	Yes	150	12:14

Figure 14
LSTM_Model1 Accuracy plot

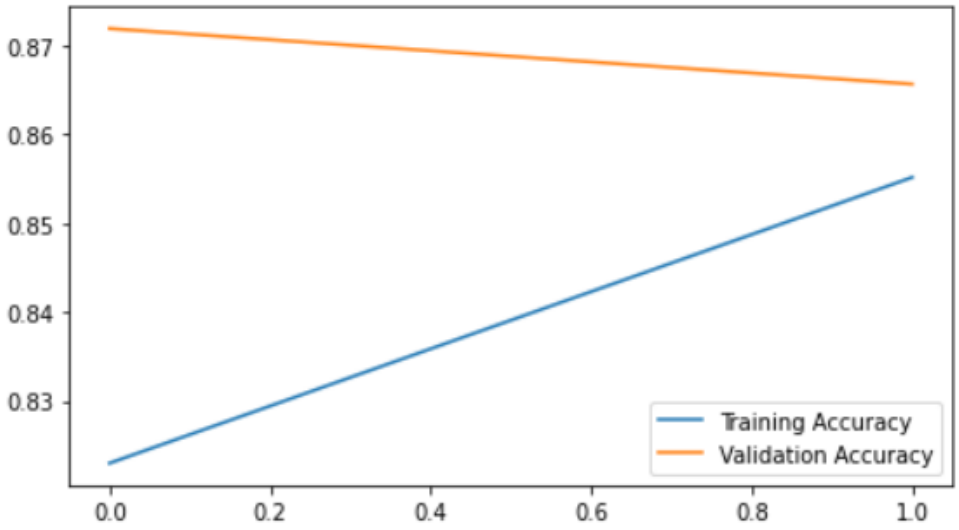


Figure 15
LSTM_Model1 Loss plot

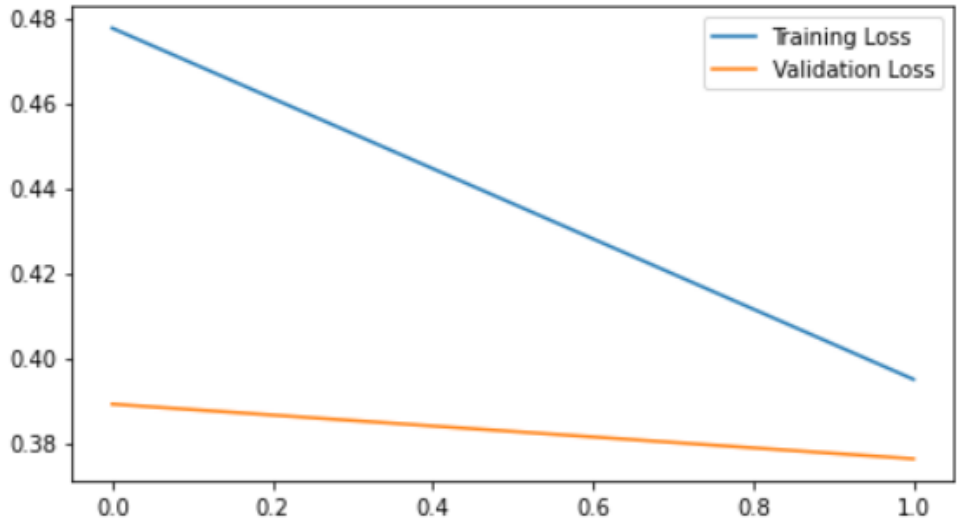


Figure 16
LSTM_Model2 Accuracy plot

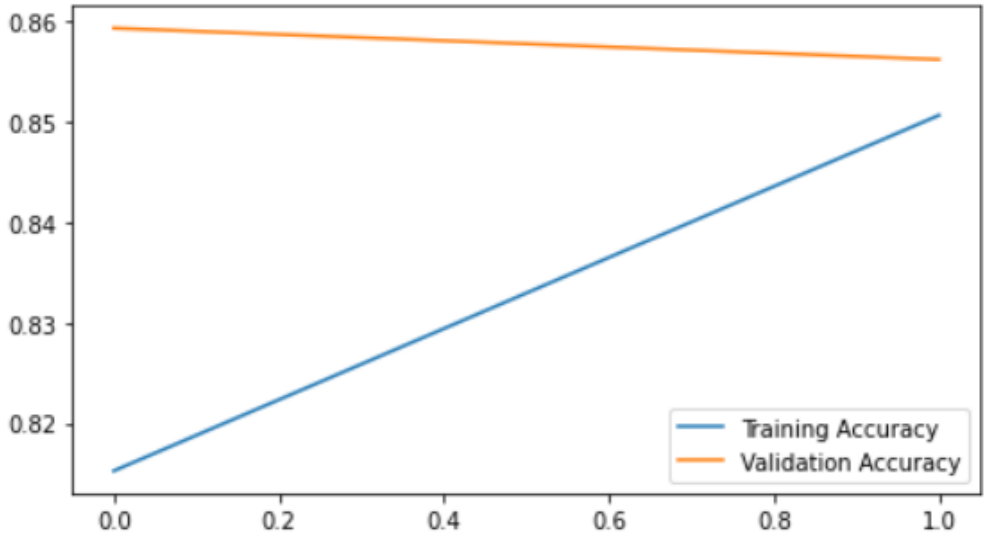


Figure 17
LSTM_Model12 Loss plot

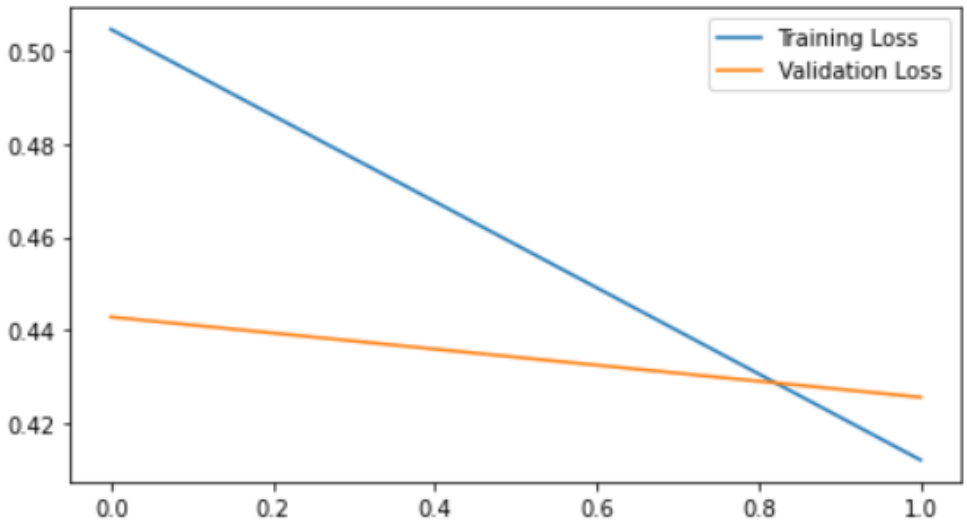


Figure 18
LSTM_Model3 Accuracy plot

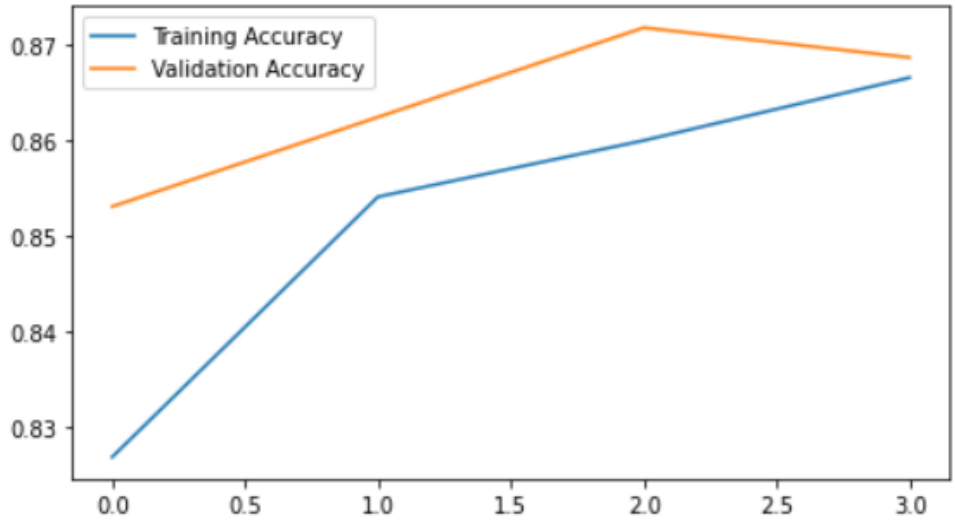


Figure 19
LSTM_Model13 Loss plot

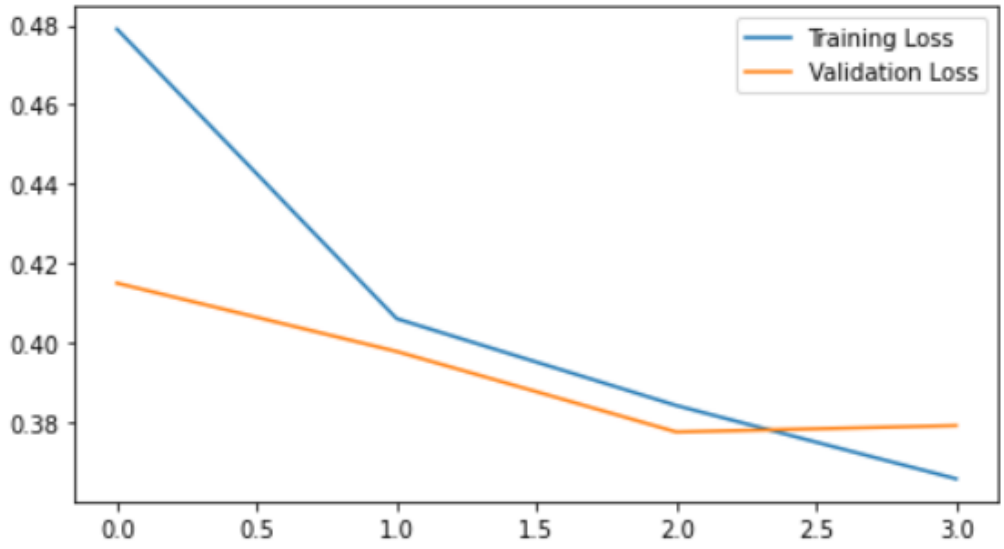


Figure 20
LSTM_Model4 Accuracy plot

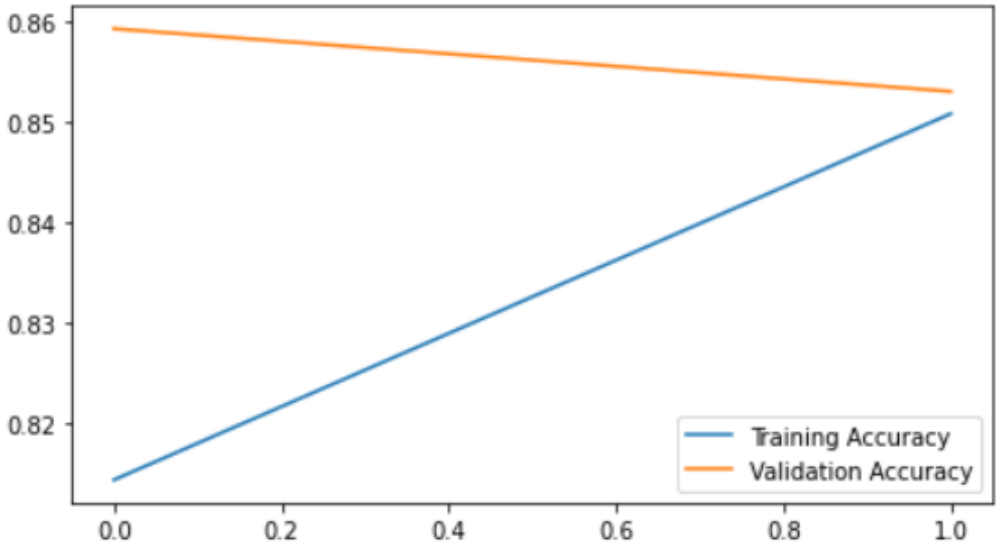


Figure 21
LSTM_Model14 Loss plot

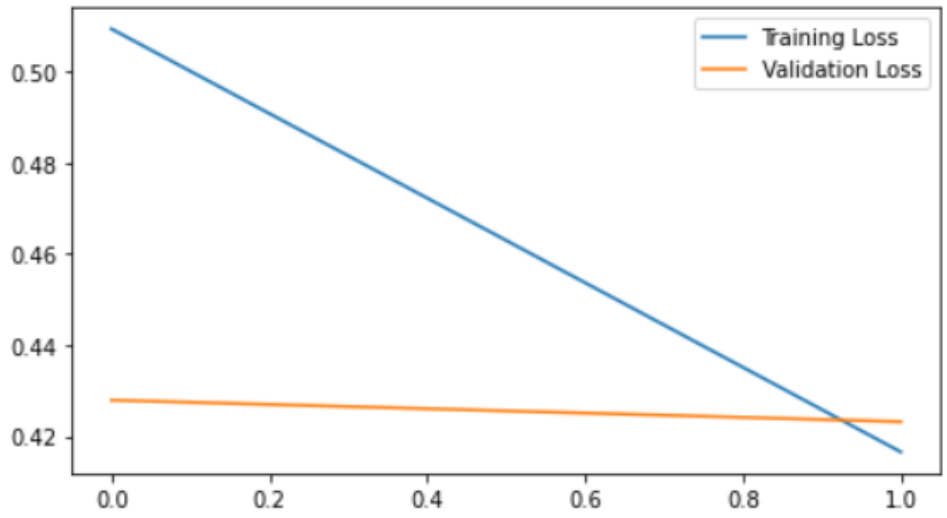


Figure 22
LSTM_Model5 Accuracy plot

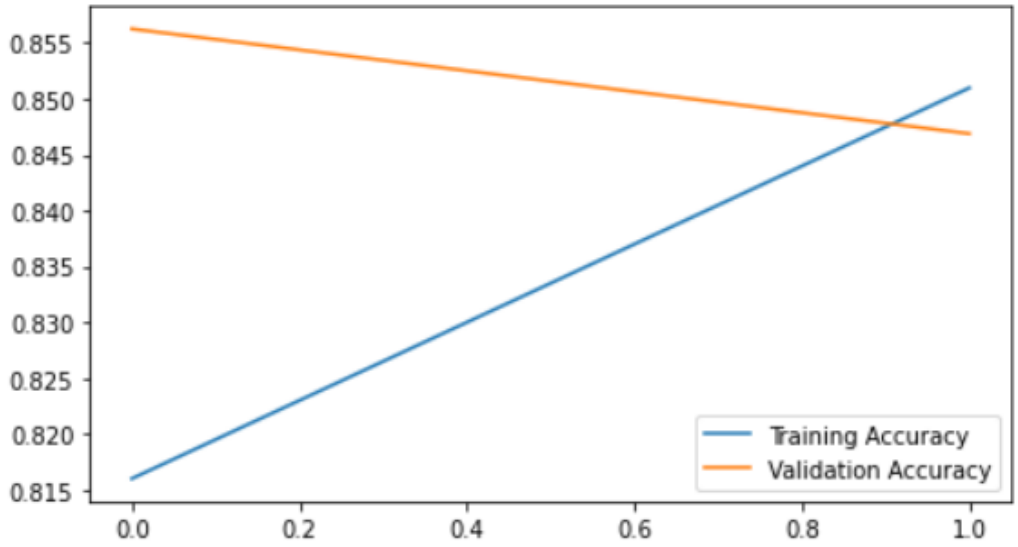


Figure 23
LSTM_Model15 Loss plot

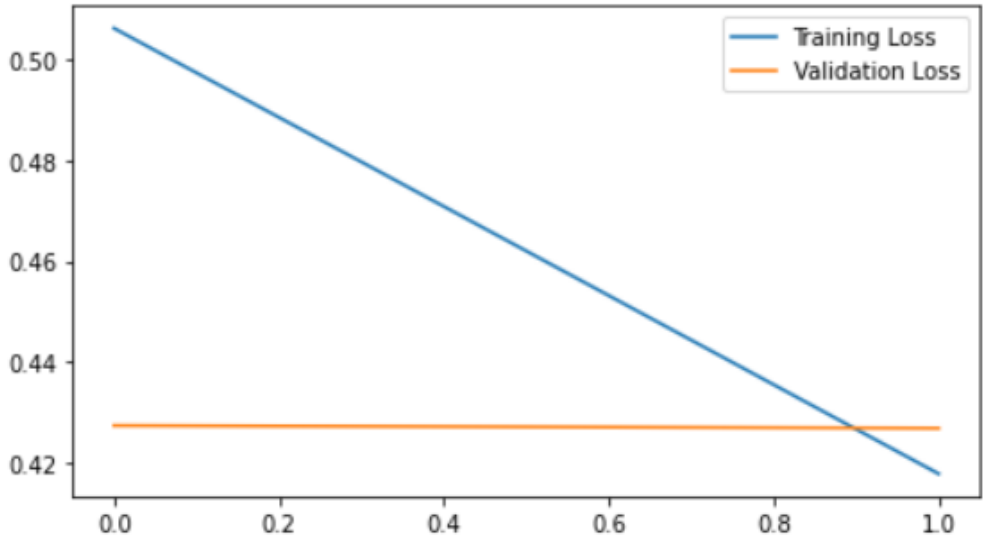


Figure 24
CNN_Model11 Accuracy plot

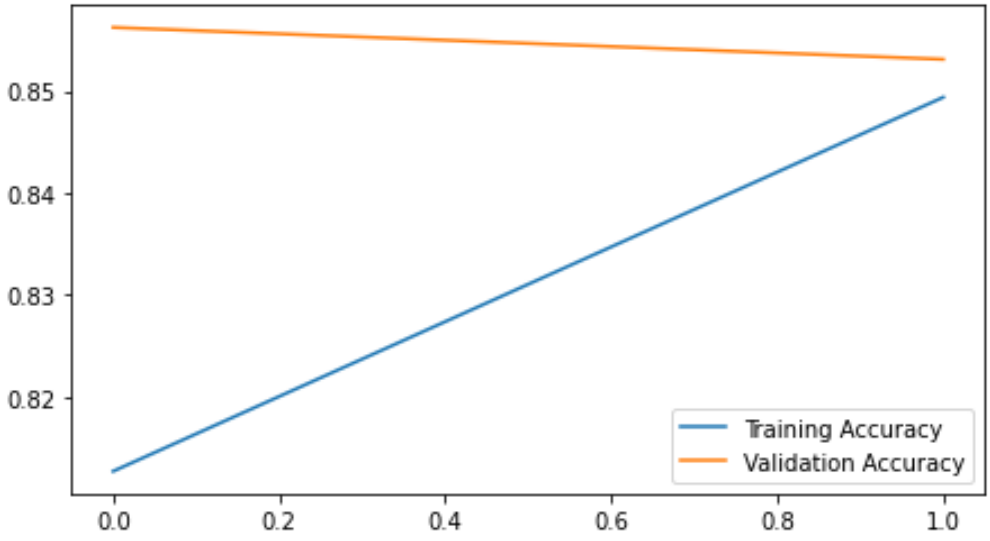


Figure 25
CNN_Model11 Loss plot

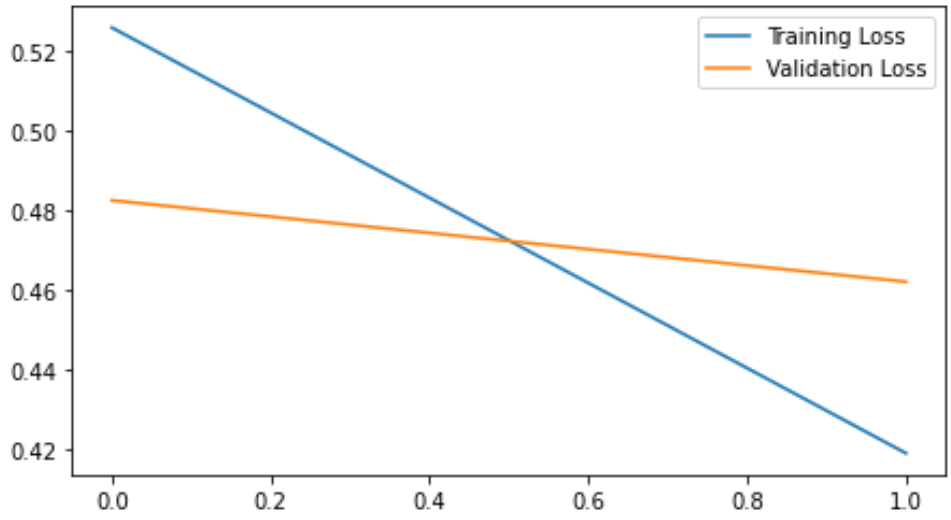


Figure 26
CNN_Model1 Accuracy and Loss Scores

Model Name	Accuracy	Loss	Direction	Regularization	Output Seqeuence Length	Process Time
CNN_model1	0.8493	0.4723	Uni	Yes	Default	00:30