Reed Ballesteros
MSDS 458-DL: Artificial Intelligence & Deep Learning
Spring 2023
Prof. Syamala Srinivasan, Ph.D.
May 14, 2023

A.3: Third Research/Programming Assignment: Project Nature & Content - Language Modeling with RNN

Abstract

Text classification plays a critical role in chatbot development such that a model can enable the system to understand and respond effectively to customers' needs. This report presents our findings from developing a Natural Language Processor (NLP) text classification model using Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs) and 1D Convolutional Neural Networks (CNNs) with the AG News dataset. Our most optimal model is an LSTM-based configuration which yields 85% accuracy in classifying the test data subset.

Introduction

Text classification plays a critical role in the development of a customer service chatbot such that a model can enable the system to understand and respond effectively to customers' needs. The model can help with recognizing intent in which the system can determine the purpose of the customer message, gather relevant information based on that purpose, and provide an accurate response in return. The model can also aid with entity extraction where the system can identify information the customer has entered into the chat such as names, addresses, product names, or order numbers. With this information the system can understand the context of the message, execute the proper request, and tailor its response to the customer. Text classification can also aid with sentiment analysis in which the chatbot can gauge the customer's mood and respond empathetically.

This report presents our findings from developing Natural Language Processor (NLP) text classification models. We built models using two frameworks: Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs) and 1D Convolutional Neural Networks (CNNs), and compared their results in classifying the AG News test dataset in terms of accuracy, loss, precision, recall, and overall F1 score.

LSTMs are used in NLP by processing a sequence of tokens by adding new content in a single memory slot, with the LSTM gate architecture controlling what content should be retained, erased, or exposed (Chang et al., 2016). Memory cells and gates in the LSTM architecture help overcome the limitations of traditional RNNs such as the vanishing gradient problem which limits the ability to retain relevant long-term information. The memory cells separate short-term and long-term memory while the hidden state in the neural network retrains immediate information. The LSTM neural network is then fed to a series of fully connected layers for text classification.

A 1D CNN is used in text classification by first transforming the input text into a sequence of word embeddings which capture the semantic meaning and contextual information of the words. The 1d convolutional layer scans the matrix of embeddings with various filters to capture local patterns, with max pooling applied to extract the most important features. It is then followed by one or more fully connected layers to be used for classification.

Literature Review

One of the prior studies of LSTM in NLP Text Classification was from the collaboration of Jianpeng Cheng, Li Dong and Mirella Lapata for the University of Edinburgh. They modified the 'vanilla' LSTM architecture to use a memory network as opposed to a memory cell (Chang et al., 2016). Doing so allows the model to store the contextual representation of each input token in their own memory slot as opposed to being contained with other information in a single cell. This allows the model to retain older and more long-term memory. The change to the memory structure resulted in better classification accuracy performance compared to base LSTM models based on sentiment analysis using the Stanford Treebank dataset and determining textual entailment using the Stanford Natural Language Inference dataset. While our LSTM-based models are not using the more sophisticated memory network architecture, the study introduces more options into modifying the LSTM framework beyond tuning hyperparameters.

One of the first studies demonstrating CNNs in text classification was published in 2014 by Yoon Kim of New York University. He created several simple CNN models and tested them against various datasets. While his baseline model did not perform well compared to other state of the art NLP models of that time, his CNN model using pre-trained word embeddings from word2vec along with hyperparameter tuning performed very well (Yoon, 2014). Our 1D CNN models do not utilize pre-trained word embeddings but perform one-hot encoding instead.

Methods

Research Design

We will conduct our research by first downloading the AG's corpus of news classification dataset and perform exploratory data analysis (EDA) on it to understand the data and its corresponding labels we will be using to build, train, and test our models. Each model is represented as an experiment. We will build and compare LSTM and 1D CNN models of varying text vectorization output sequence lengths (None, 96) and dropout regularization levels (none, 0.2, 0.3). For each model we will record their accuracy and loss against the test subset, as well as precision, recall, F1 score, and RMSE. We will also plot each model's training and validation loss and accuracy over their respective number of epochs of training. Training for each model will automatically stop after validation accuracy does not improve after a given number of epochs, in which the best training epoch will be saved before the model overfits to the training data. Confusion matrices for each model will also be charted to understand how well they perform for each image classification.

Implementation

Our models will be created using the Keras and TensorFlow frameworks available in Python. These are the most common tools used in developing such models based on their ease of use, customization options available, and their scalability for each LSTM, convolutional, or dense layer. The embedding layer for the LSTM models will use various floating-point vectors of various dimension sizes (64, 128, 256) with masking enabled, while the embedding layer for the 1D convolution models will use one-hot encoding. The convolutional layer will use the ReLU activation function as its simplicity in calculation can make the feedforward and backpropagation process fast, which will be helpful when we train deep models that contain a large number of parameters. The final 4-way (based on the number of news categories in the AG News dataset) classification output layer will use a softmax activation function for both LSTM and 1D convolution models. We will also build models set to various text vectorization output sequence lengths (20, 30, 70, 96) to find the most optimal value.

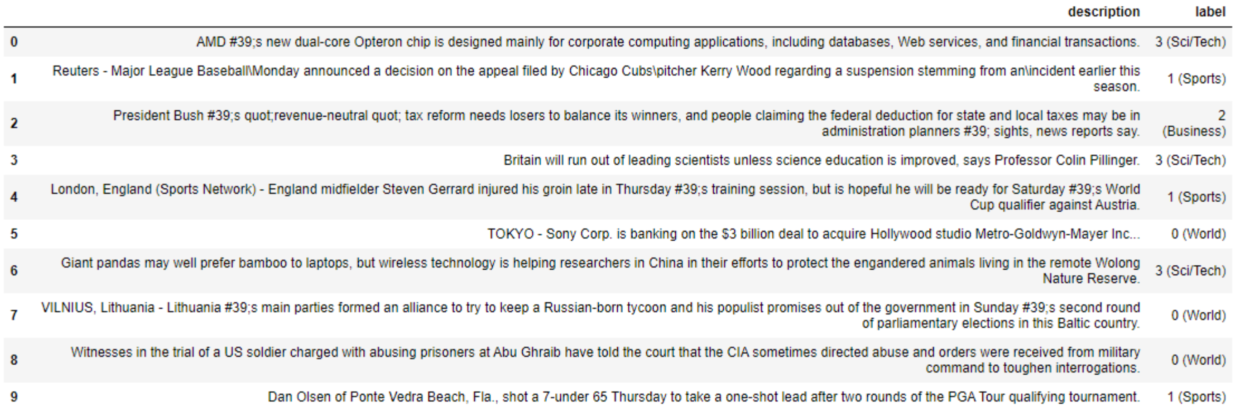
The Keras model compiler will be configured to use the RMSProp optimizer, calculate cross entropy loss using the SparseCategorialCrossentropy class, and model performance will be based on accuracy metrics.

Dataset: AG News Classification Dataset

The AG's Corpus of news articles is a collection of more than 1 million news articles initially provided for use in the academy community for areas of research such as data mining, information retrieval, and data compression. A subset of the collection, the AG's News Topic Classification dataset, is commonly used as a benchmark dataset for NLP and machine learning research. Each news article in the dataset is labeled with one of four predefined categories: World, Sports, Business, and Sci/Tech. The classification labels allow researchers to train and evaluate machine learning models on the dataset for tasks such as topic classification, sentiment analysis, or text categorization. The models we will develop for our experiments will use this categorized dataset as our benchmark, split into training, validation, and test sets.

Exploratory Data Analysis (EDA)

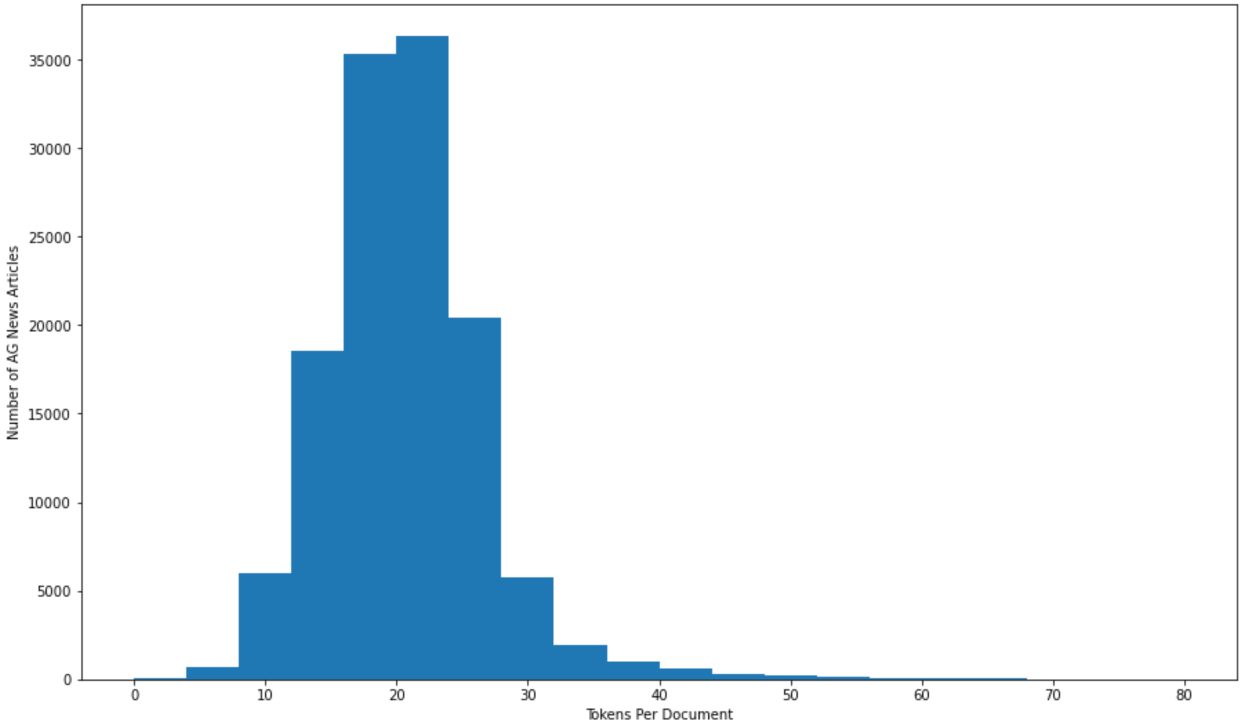
An initial exploratory data analysis (EDA) on the AG's News Topic Classification dataset reveals that it contains 127600 total articles, curated with an equal distribution of 31900 articles in each category (0: World, 1: Sports, 2: Business, 3: Sci/Tech). A sample of the AG's News content is shown below.



	description	label
0	AMD #39;s new dual-core Opteron chip is designed mainly for corporate computing applications, including databases, Web services, and financial transactions.	3 (Sci/Tech)
1	Reuters - Major League Baseball\Monday announced a decision on the appeal filed by Chicago Cubs\pitcher Kerry Wood regarding a suspension stemming from an\incident earlier this season.	1 (Sports)
2	President Bush #39;s quot;revenue-neutral quot; tax reform needs losers to balance its winners, and people claiming the federal deduction for state and local taxes may be in administration planners #39; sights, news reports say.	2 (Business)
3	Britain will run out of leading scientists unless science education is improved, says Professor Colin Pillinger.	3 (Sci/Tech)
4	London, England (Sports Network) - England midfielder Steven Gerrard injured his groin late in Thursday #39;s training session, but is hopeful he will be ready for Saturday #39;s World Cup qualifier against Austria.	1 (Sports)
5	TOKYO - Sony Corp. is banking on the $3 billion deal to acquire Hollywood studio Metro-Goldwyn-Mayer Inc...	0 (World)
6	Giant pandas may well prefer bamboo to laptops, but wireless technology is helping researchers in China in their efforts to protect the engandered animals living in the remote Wolong Nature Reserve.	3 (Sci/Tech)
7	VILNIUS, Lithuania - Lithuania #39;s main parties formed an alliance to try to keep a Russian-born tycoon and his populist promises out of the government in Sunday #39;s second round of parliamentary elections in this Baltic country.	0 (World)
8	Witnesses in the trial of a US soldier charged with abusing prisoners at Abu Ghraib have told the court that the CIA sometimes directed abuse and orders were received from military command to toughen interrogations.	0 (World)
9	Dan Olsen of Ponte Vedra Beach, Fla., shot a 7-under 65 Thursday to take a one-shot lead after two rounds of the PGA Tour qualifying tournament.	1 (Sports)

Before we use the dataset for model training and evaluation, we need to preprocess the data to turn all letters into lowercase and remove all punctuations. Doing so will standardize words and reduce the overall total vocabulary as word variations due to letter case and punctuation would count as unique works and make the training process unnecessarily more complex. We will also edit the dataset to remove common words such as 'the,' 'and,' or 'but', etc. that do not add significant context or semantics to the documents. The list of words to be deleted is based on the set of 40 stop words provided by the Natural Language Toolkit (NLTK).

After data processing, we found 2579419 words in the 127600 articles of the dataset. Each article ranges from 2 to 95 tokens. A distribution of the number of tokens per document is shown below.



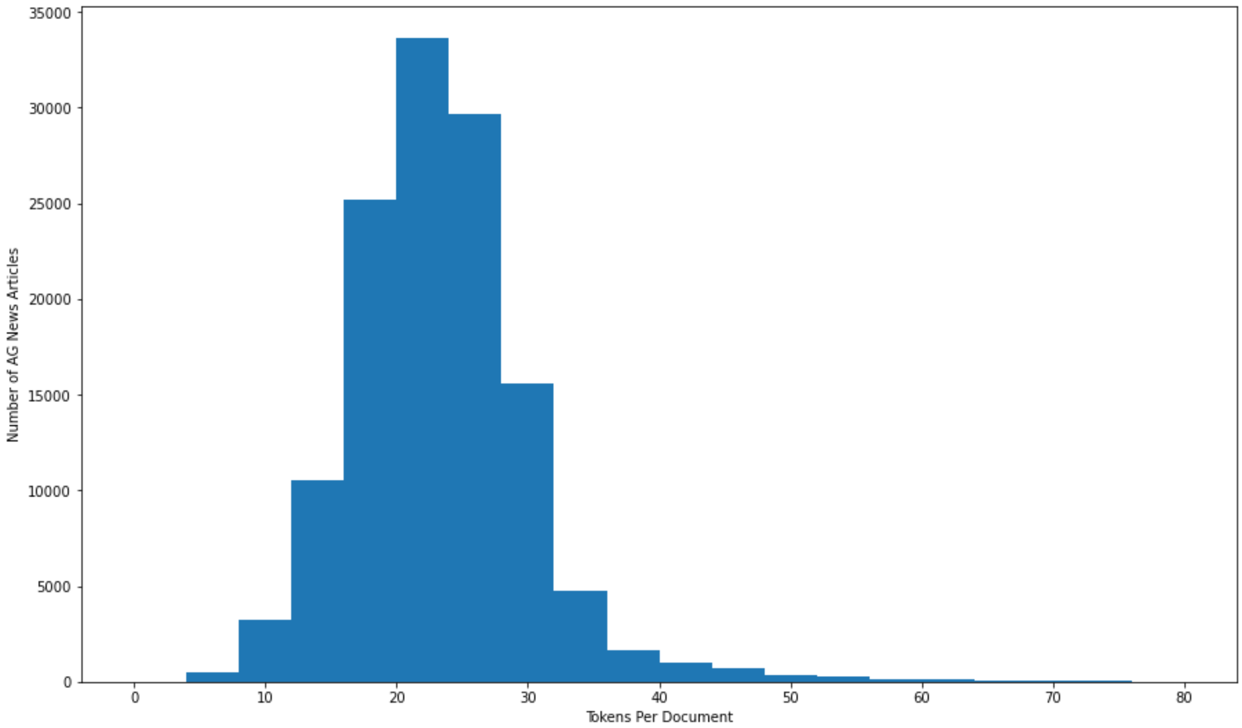
While most documents lie in the 20 token range, the distribution is right-skewed to reflect the presence of some longer articles in the dataset.

Editing Stop Words: Removing Articles

We also performed a variation on preprocessing data in which we removed article words (‘the,’ ‘a,’ ‘an’) from the NLTK stop words list. A table comparing the differences in distribution is below.

STOPWORDS	total words	total news articles	min tokens	max tokens	total vocabulary
Standard	2579419	127600	2	95	95827
the', 'a', 'an' removed	2915636	127600	2	107	95830

With article words removed, the total number of words, max tokens, and vocabulary has increased. The distribution tokens per document based on this stop words list is below.



Due to the presence of articles in each document in the dataset we see more articles lying in the 30-token range.

The large presence of articles could unnecessarily impact the training process of our models, especially since those words do not provide significant context or semantics in categorizing articles. Going forward, it is best to preprocess the data by removing words from the list of NLTK stop words as-is, without the removal of articles.

Text Vectorization - Changing Vocabulary Size

While preprocessing the data we also wanted to observe if changing the maximum vocabulary size in text vectorization would have an effect on the number of total words and max tokens. A table of variable vocabulary sizes is below.

max_tokens	total words	total news articles	min tokens	max tokens	total vocabulary
None (default)	2579419	127600	2	95	95827
60000	2579419	127600	2	95	60000
30000	2579419	127600	2	95	30000

We find that changing the vocabulary size did not make any change to the number of max tokens in the dataset. We also found the distribution of the number of tokens per document did not change. Going forward, we will not limit the vocabulary size in text vectorization for our models.

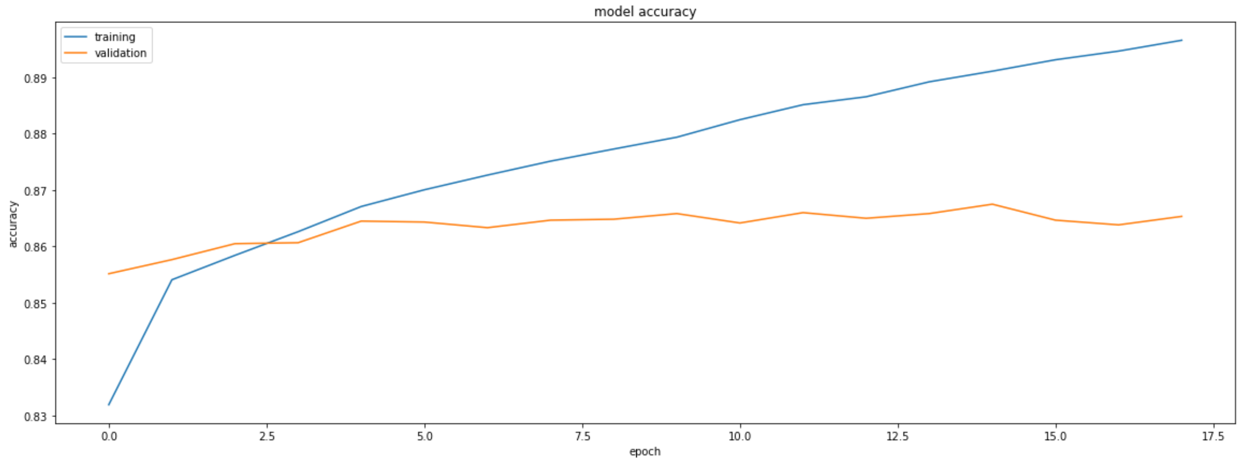
Text Vectorization - Changing Output Sequence Length

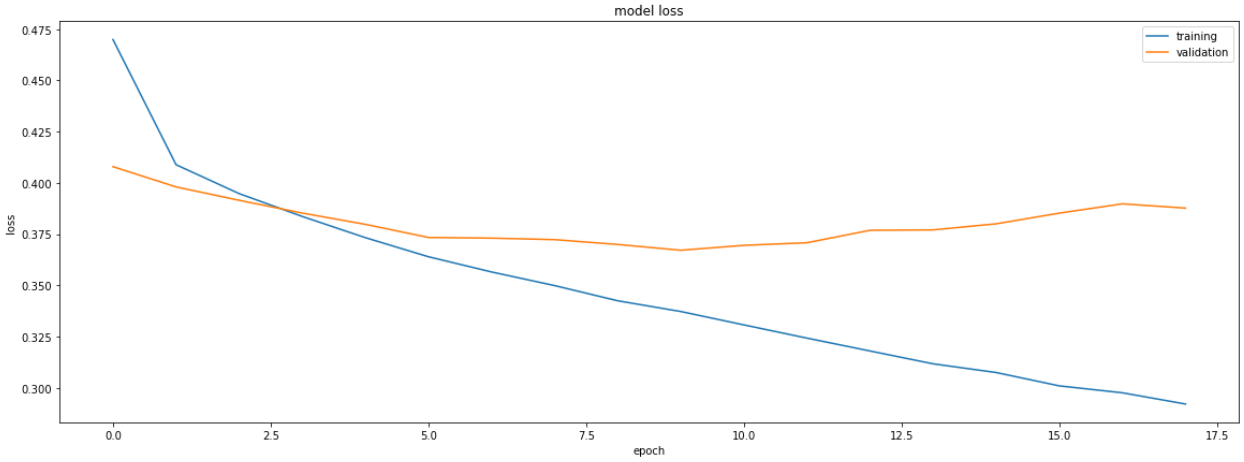
We will create various LSTM and 1D Convolutional models using two kinds of output sequence lengths: None (default) and 96. In other words, given a set number of tokens, each article will be preprocessed to that number of tokens. Given that the max number of tokens in the dataset is 95, all articles in the dataset will be padded to 96 tokens. That being said, while most articles contain about 20 tokens, many articles will contain more padding than actual content. We will compare models trained using padded articles against those without.

Results

Experiment 1 – LSTM, Output Sequence Length None, Embedding Dimension 64, Dropout 0.2

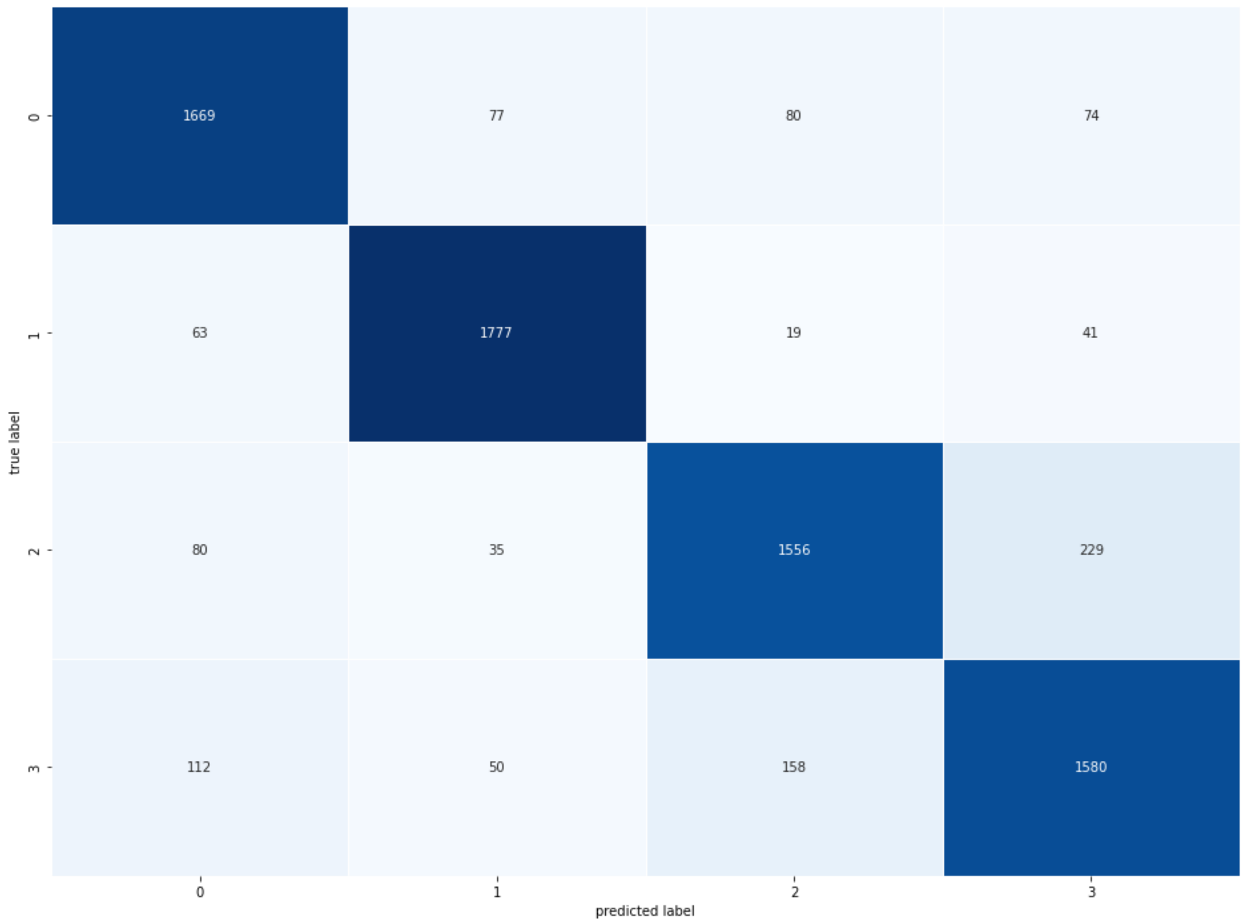
Our first experiment is an LSTM-based model using no set output sequence length embedded to 64-dimension word vectors. We chart the training and validation accuracy and cross entropy loss below.





From the charts above, we see validation results start to diverge from the training results after just 3 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 86.5%. We can use this as a baseline for all other LSTM-based experiments that follow.

The experiment yields the following confusion matrix:



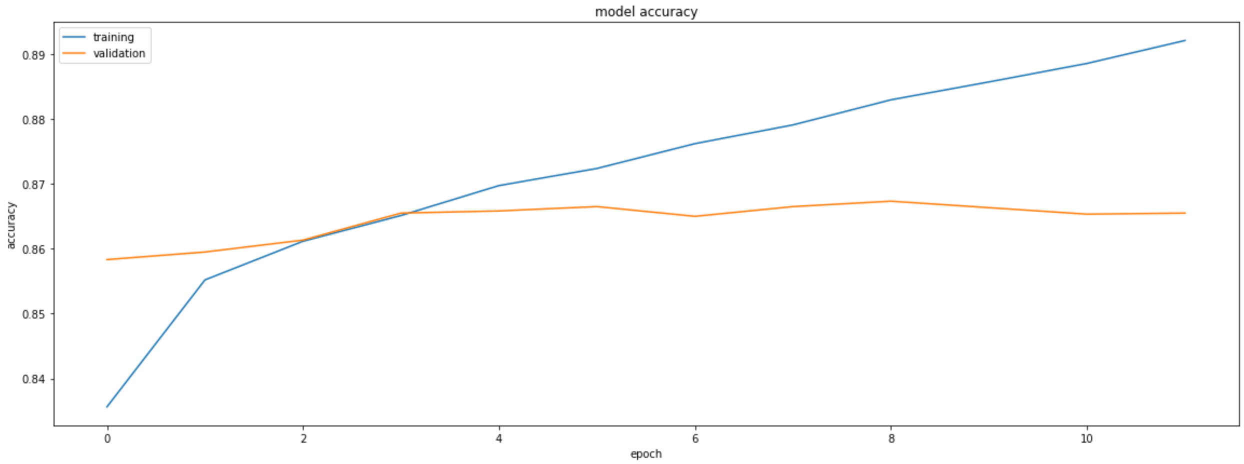
he numbered labels correspond to the following classification in the AG News dataset:

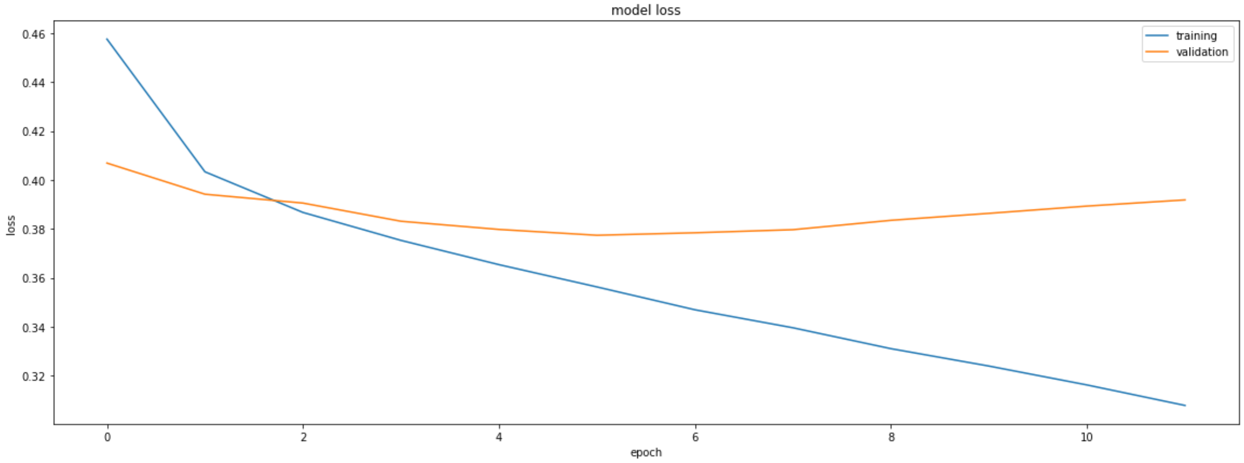
Label	**Classification**
0	World
1	Sports
2	Business
3	Sci/Tech

The confusion matrix above shows that while the model can classify most articles to their respective classes, it does have some difficulty correctly categorizing articles between business and science/technology. This could be due to the technical and statistical nature of articles that could be common between both subjects. We will see if this issue continues in other experiments.

Experiment 2 – LSTM, Output Sequence Length None, Embedding Dimension 128, Dropout 0.2

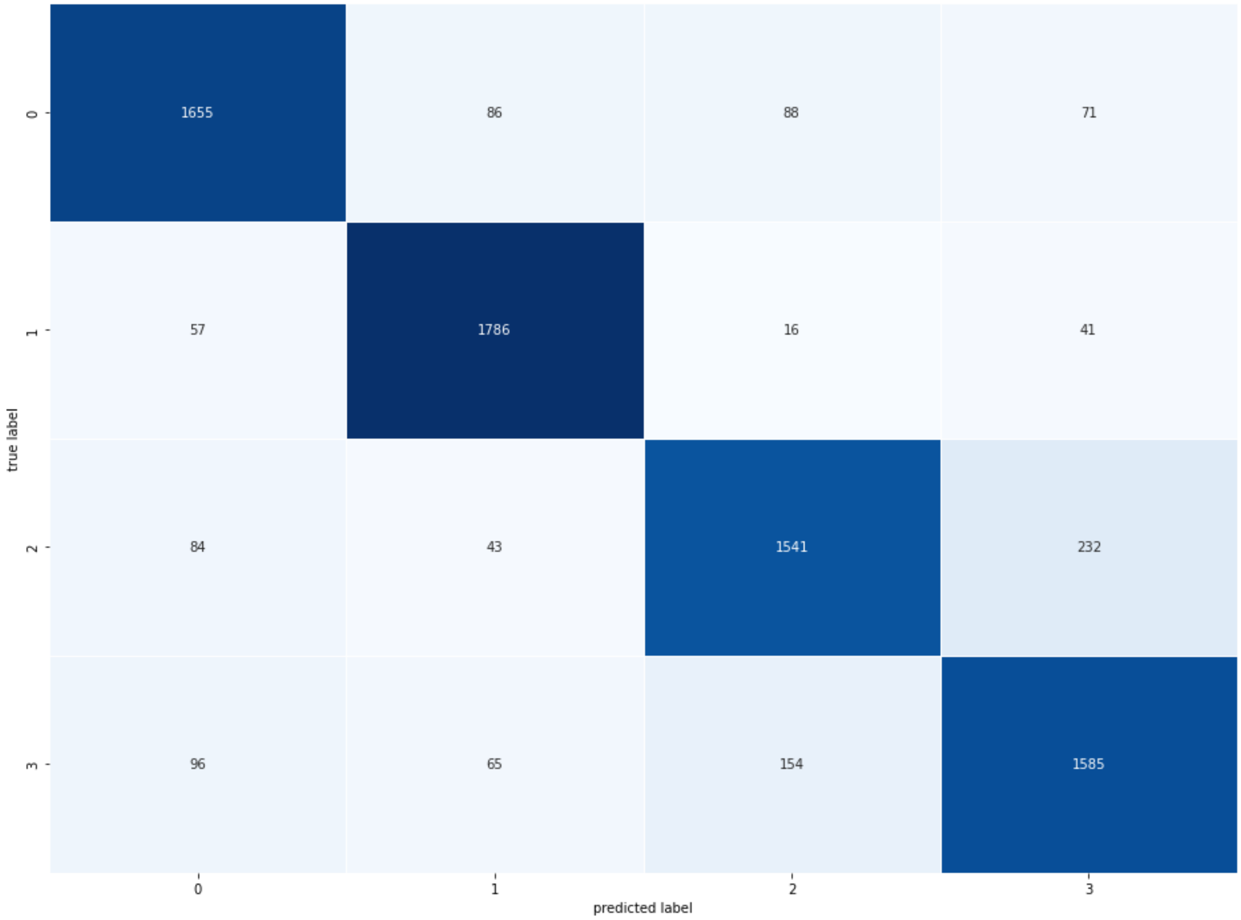
This experiment is an LSTM model using no set output sequence length embedded to 128-dimension word vectors. We chart the training and validation accuracy and cross entropy loss below.





From the charts above, we see validation results start to diverge from the training results after just 3 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 86.4%, marginally worse than what we've seen in experiment 1.

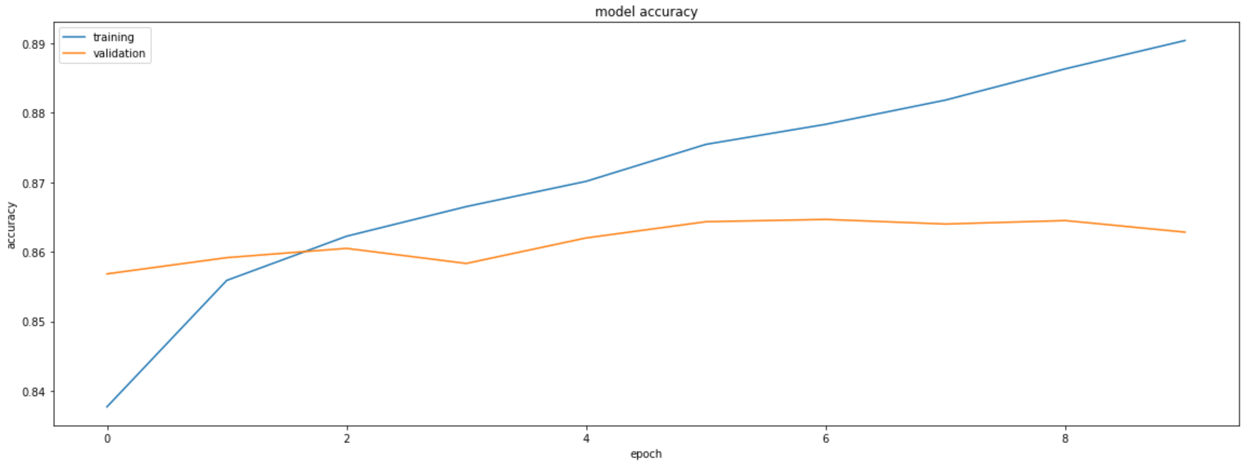
The experiment yields the following confusion matrix:

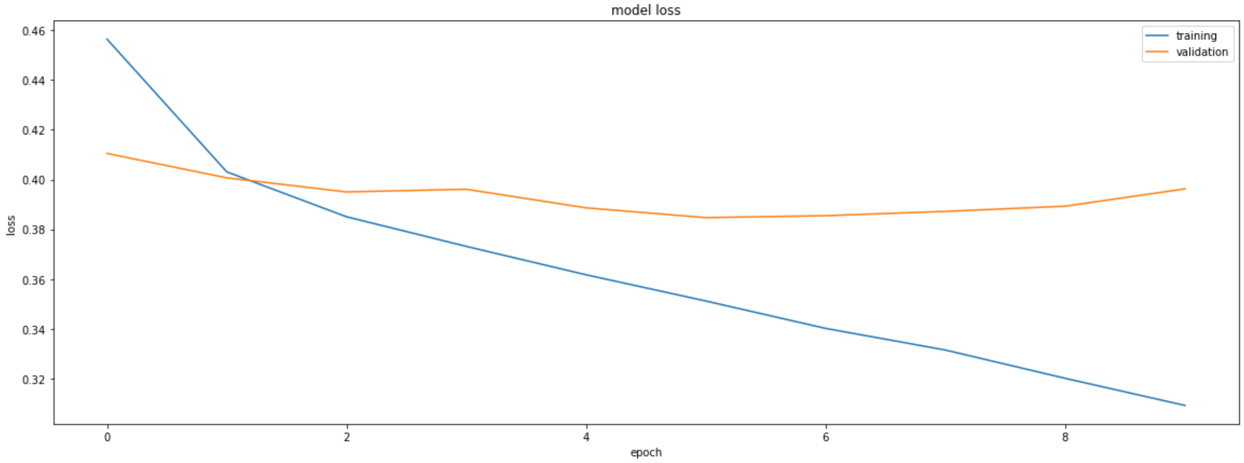


The confusion matrix above shows this model also has some difficulty correctly categorizing articles between business and science/technology.

Experiment 3 – LSTM, Output Sequence Length None, Embedding Dimension 256, Dropout 0.2

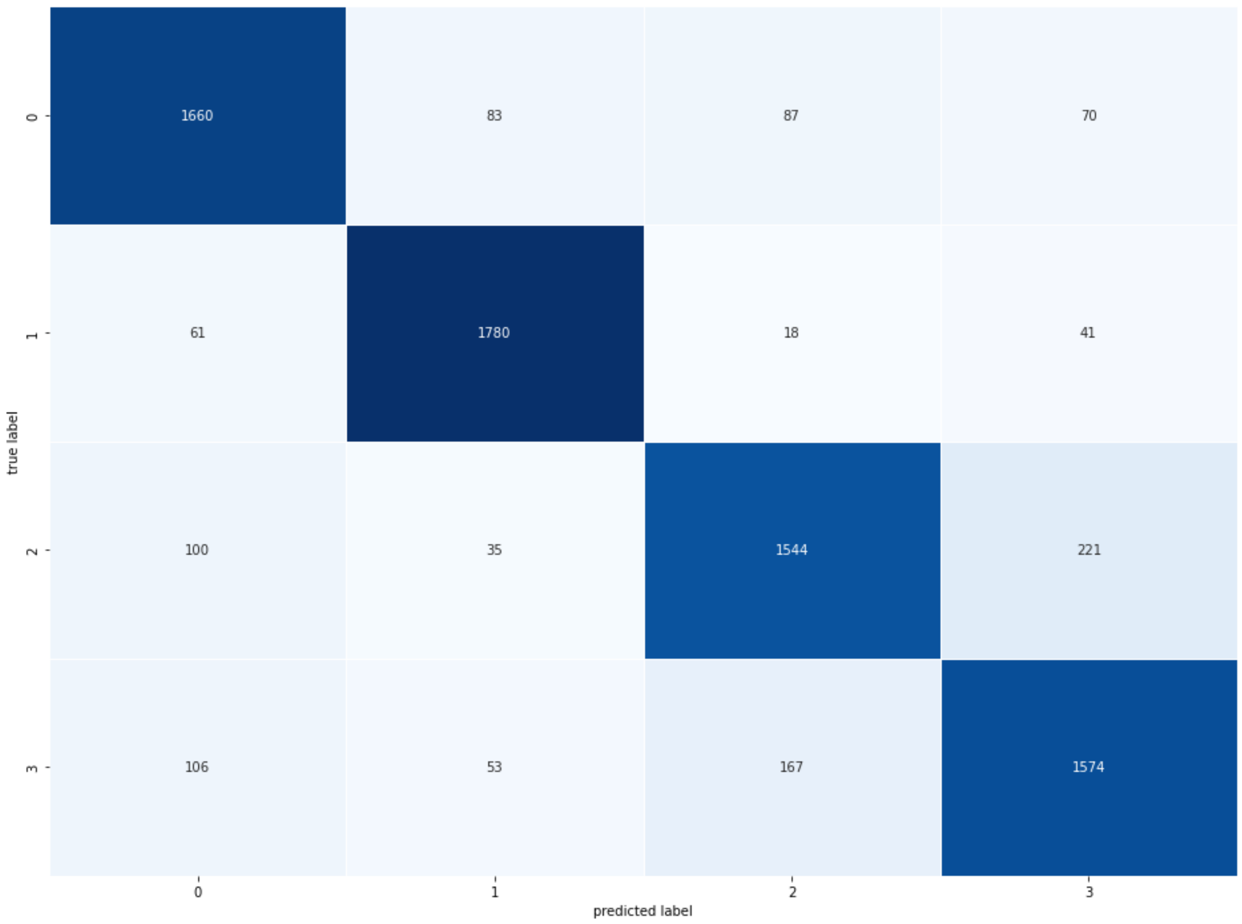
We created an LSTM model using no set output sequence length embedded to 256-dimension word vectors. We chart the training and validation accuracy and cross entropy loss below.





From the charts above, we see validation results start to diverge from the training results after just 3 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 85.9%, .5% worse compared to experiment 2. We start to wonder if we've found the most optimal embedding dimension at 64 in experiment 1.

The experiment yields the following confusion matrix:

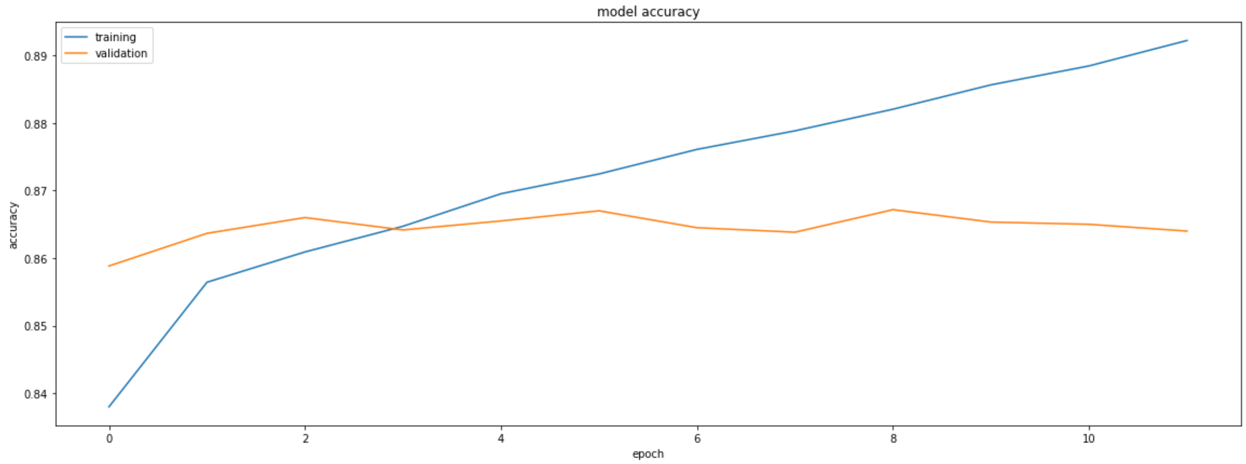


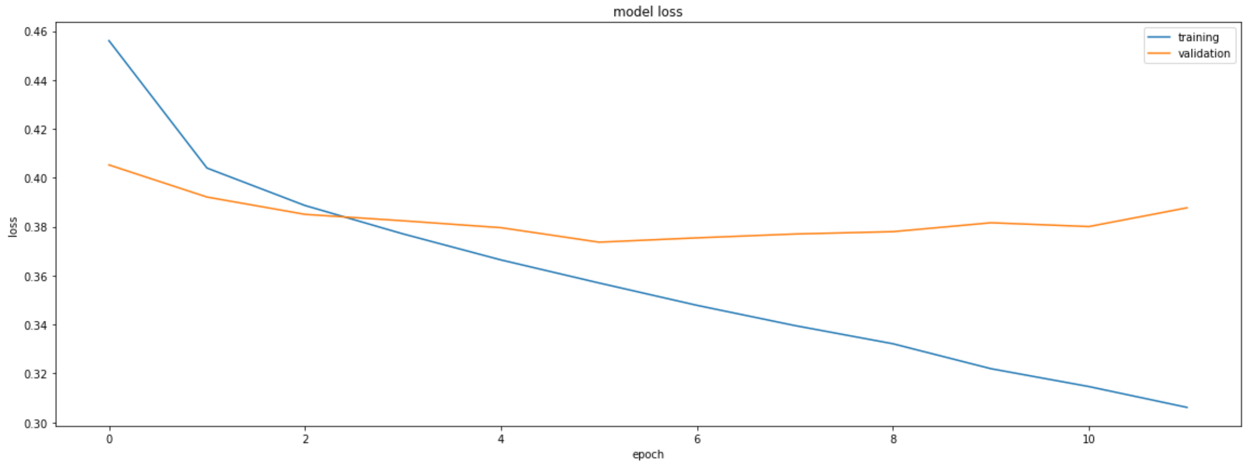
As demonstrated in previous models, the confusion matrix above shows this model has some difficulty correctly categorizing articles between business and science/technology.

Experiments 4 to 6 will utilize padding all articles in the dataset to 96 tokens via text vectorization.

Experiment 4 – LSTM, Output Sequence Length 96, Embedding Dimension 64, Dropout 0.2

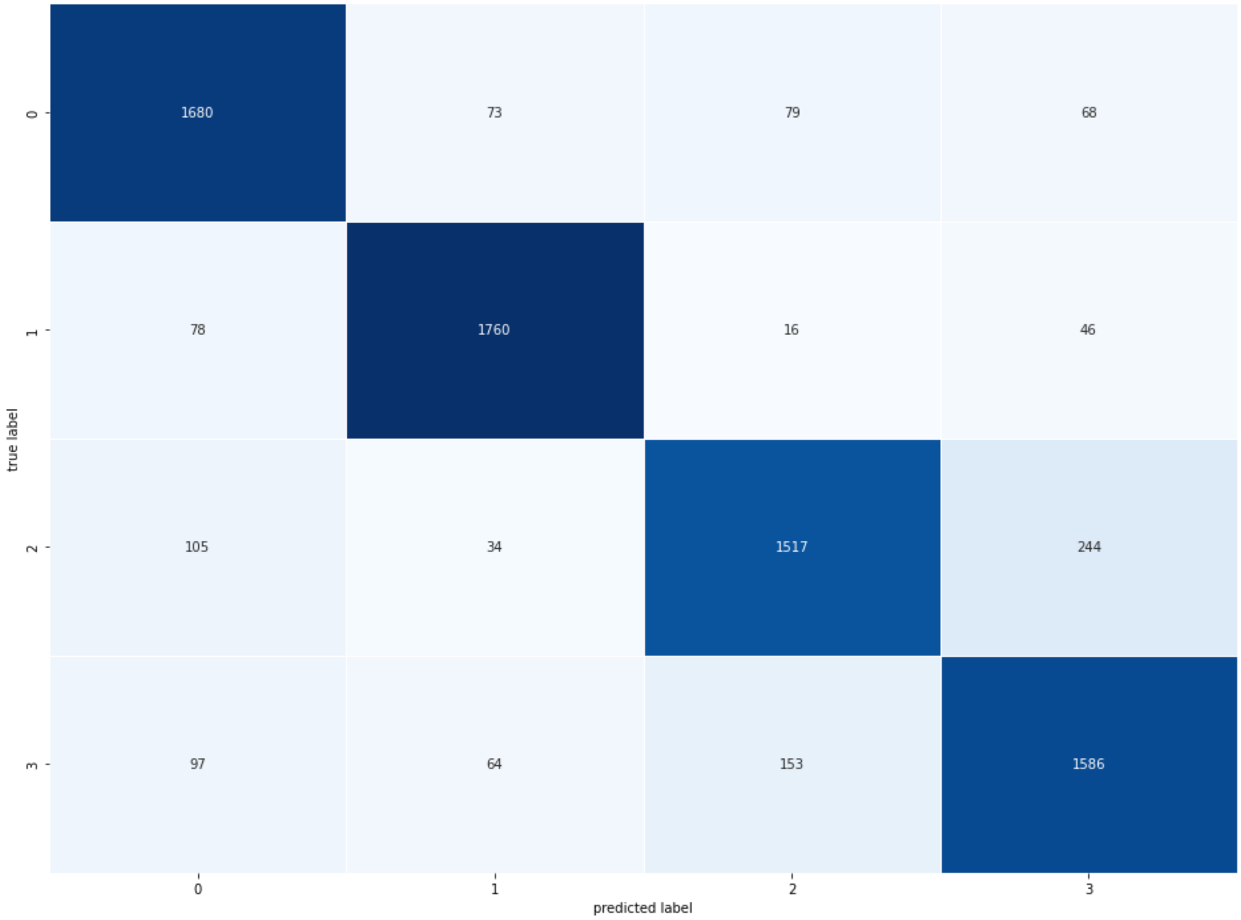
Experiment 4 is an LSTM model which uses an output sequence length of 96, thus padding all articles in the dataset to 96 tokens. We embed the articles to 64-dimension word vectors. We chart the training and validation accuracy and cross entropy loss below.





We see validation results start to diverge from the training results after just 3 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 85.9%, 0.5% worse than what we've seen in experiment 1 at the same embedding dimension, and the same accuracy as with experiment 3, which was the worst-performing model so far. We will see if increasing the embedding dimensions in experiment 5 yields better or worse results.

The experiment yields the following confusion matrix:

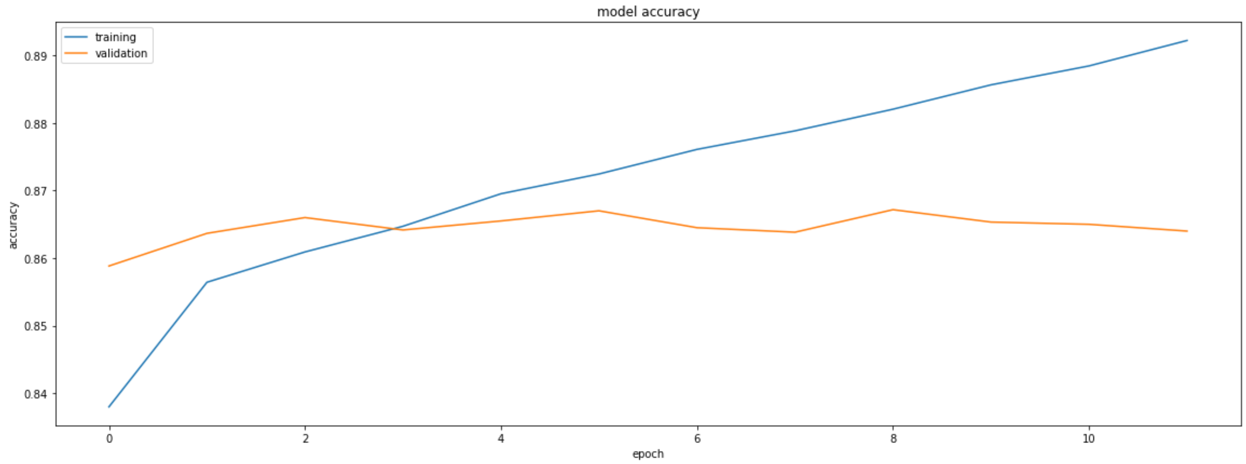


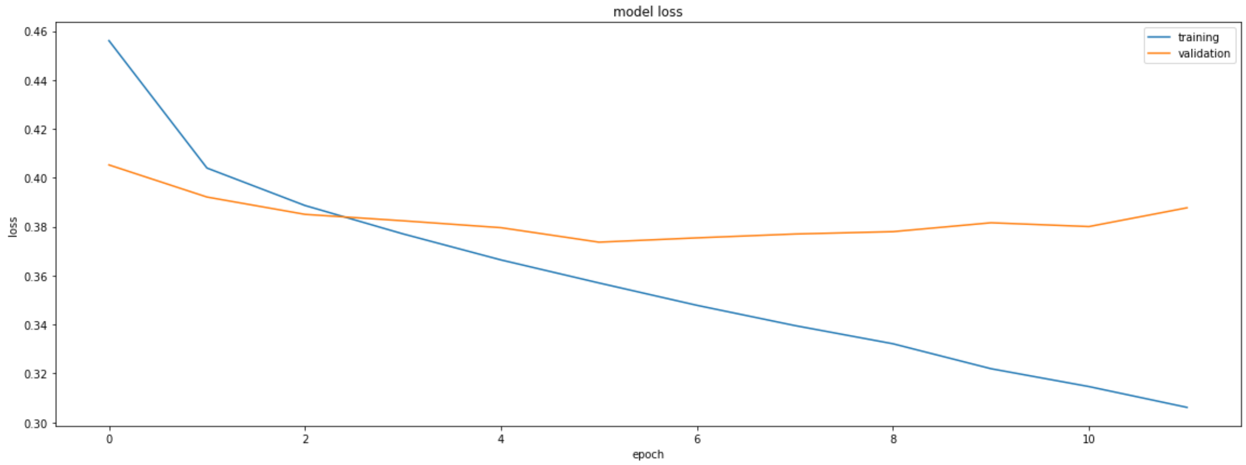
	0	1	2	3
0	1680	73	79	68
1	78	1760	16	46
2	105	34	1517	244
3	97	64	153	1586

Padding articles to 96 tokens does not improve the mis-categorization between business and science/technology articles.

Experiment 5 – LSTM, Output Sequence Length 96, Embedding Dimension 128, Dropout 0.2

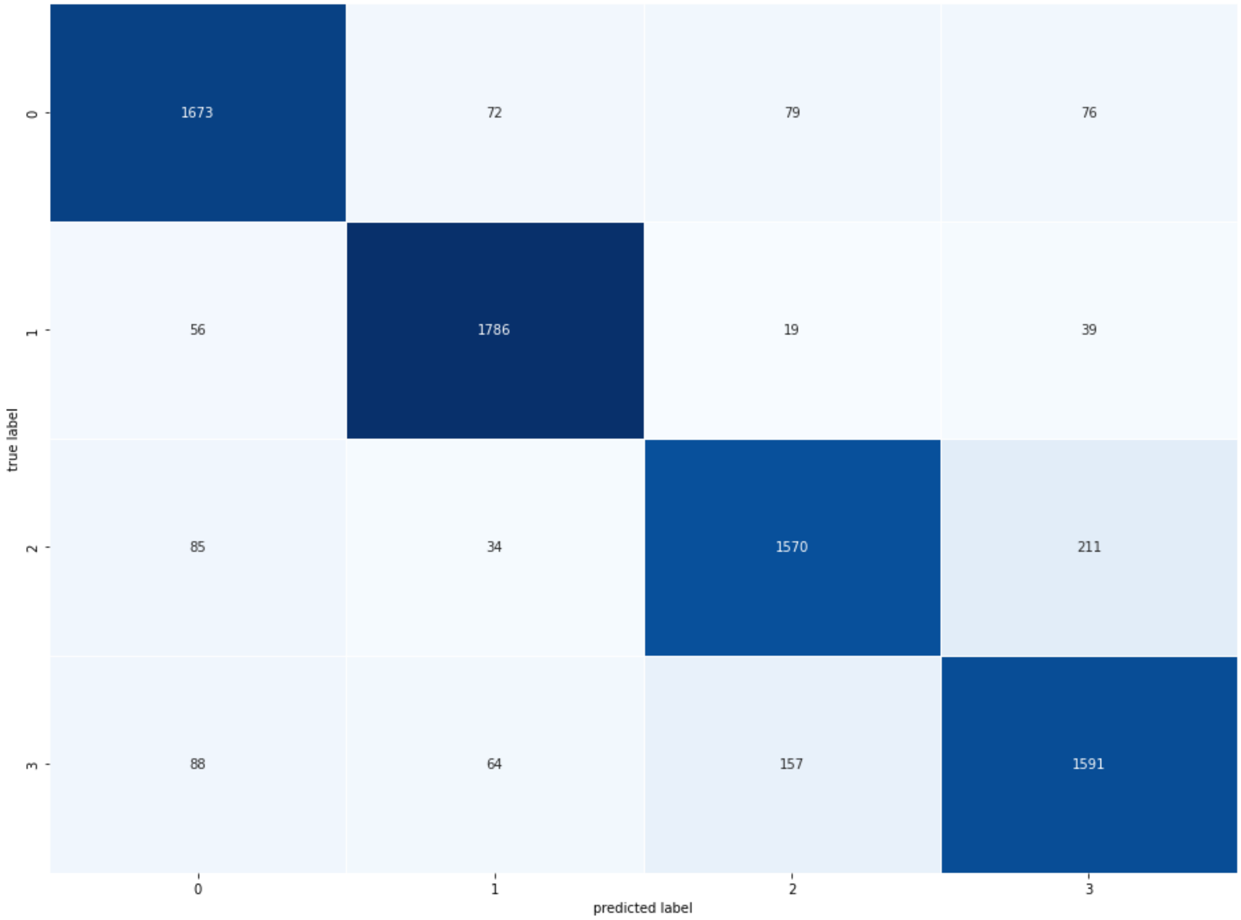
This is an LSTM model with an output sequence length of 96 in which we embed the articles to 128-dimension word vectors. We chart the training and validation accuracy and cross entropy loss below.





As in previous models, validation results start to diverge from the training results after just 3 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 87.1%, our best-performing model so far. We will see if increasing the embedding dimensions in experiment 6 can continue the trend.

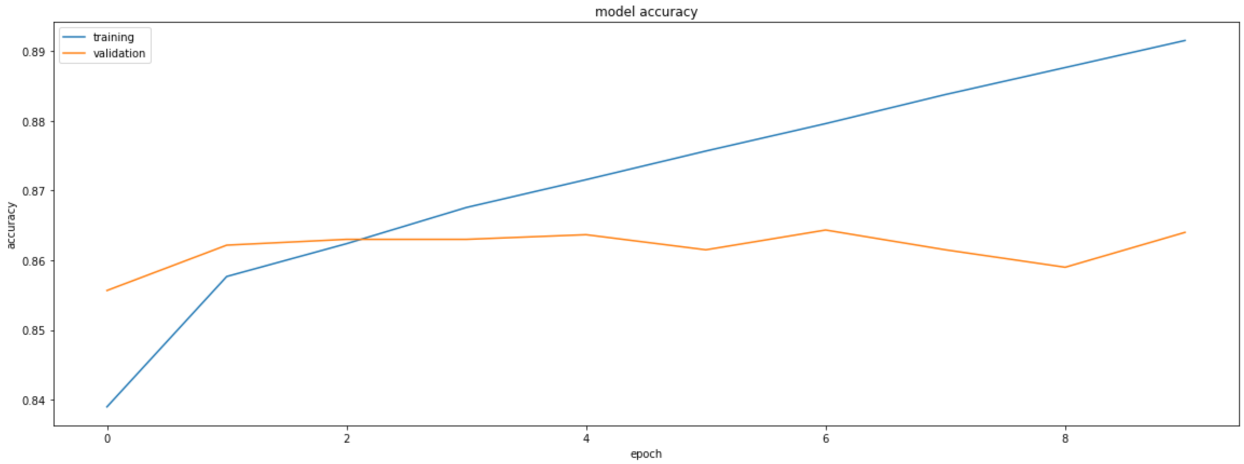
The experiment yields the following confusion matrix:

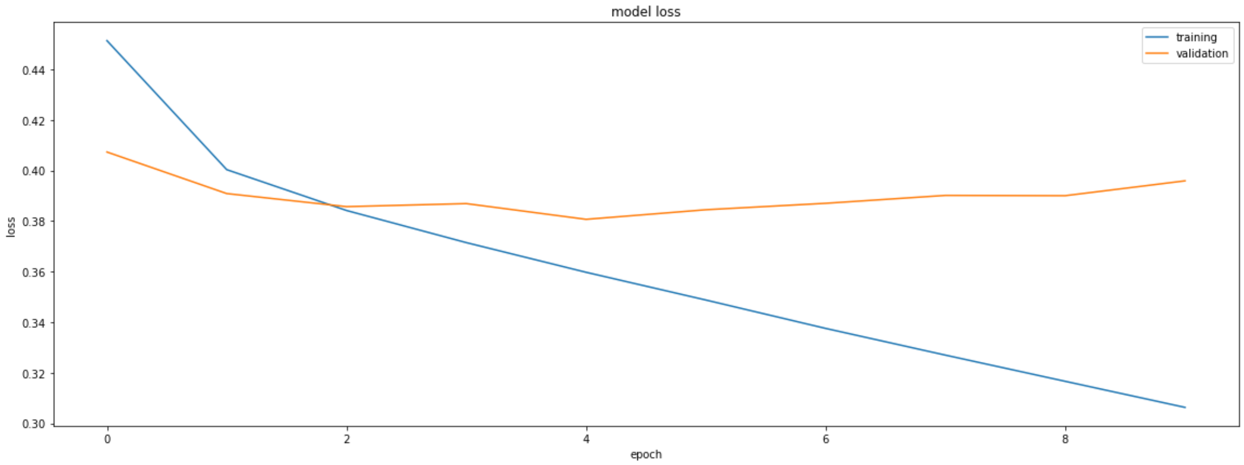


As demonstrated in previous models, the confusion matrix above shows this model has some difficulty correctly categorizing articles between business and science/technology.

Experiment 6 – LSTM, Output Sequence Length 96, Embedding Dimension 256, Dropout 0.2

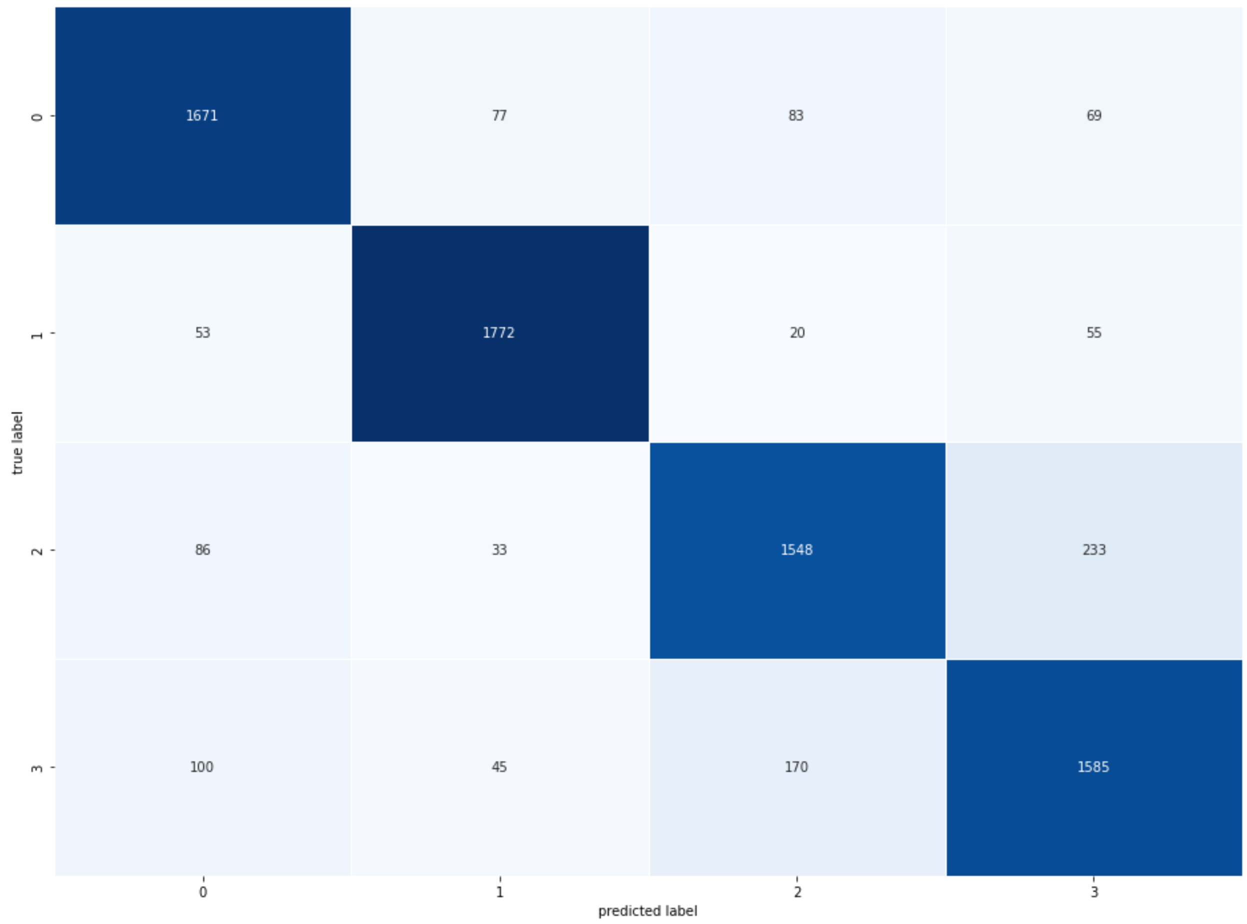
This experiment uses an LSTM model with an output sequence length of 96 and embeds the articles to 128-dimension word vectors. We chart the training and validation accuracy and cross entropy loss below.





Validation results start to diverge from the training results after just 3 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 86.4%. We find that embedding the padded to 128-dimension word vectors yielded the best accuracy against the AG News test dataset, and increasing the dimensions started to lead to worse results.

The experiment yields the following confusion matrix:



Despite tuning the output sequence length and embedding dimension, some difficulty categorizing business and science/technology models exist between all LSTM models developed so far.

A table comparing LSTM models based on output sequence length and embedding dimensions is below. We found that padding all documents in the dataset to 96 tokens while embedding them to 128-dimension word vectors yielded the best accuracy against the AG News test dataset with 87.1%.

LSTM - Embedding Dimensions

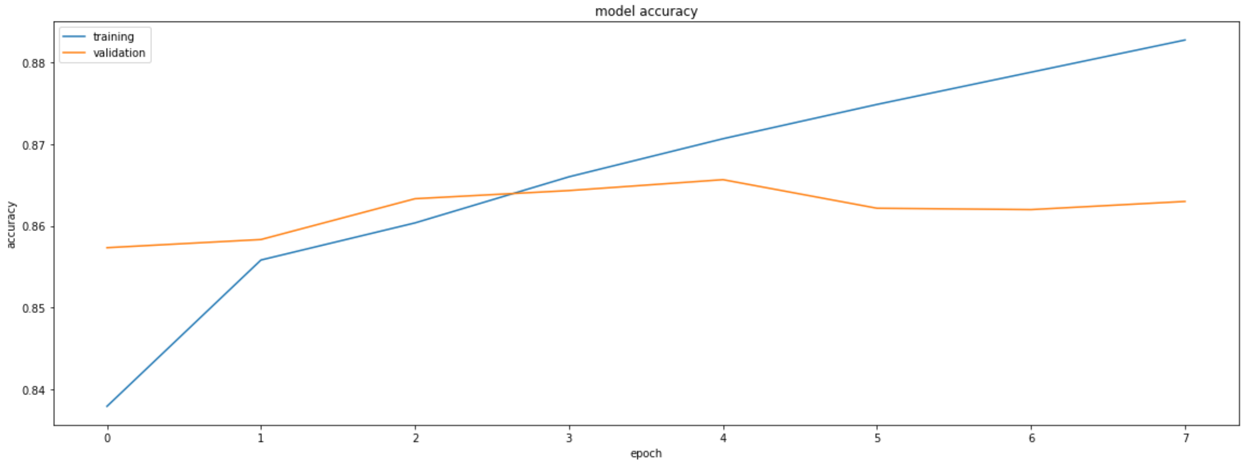
Dropout: 0.2

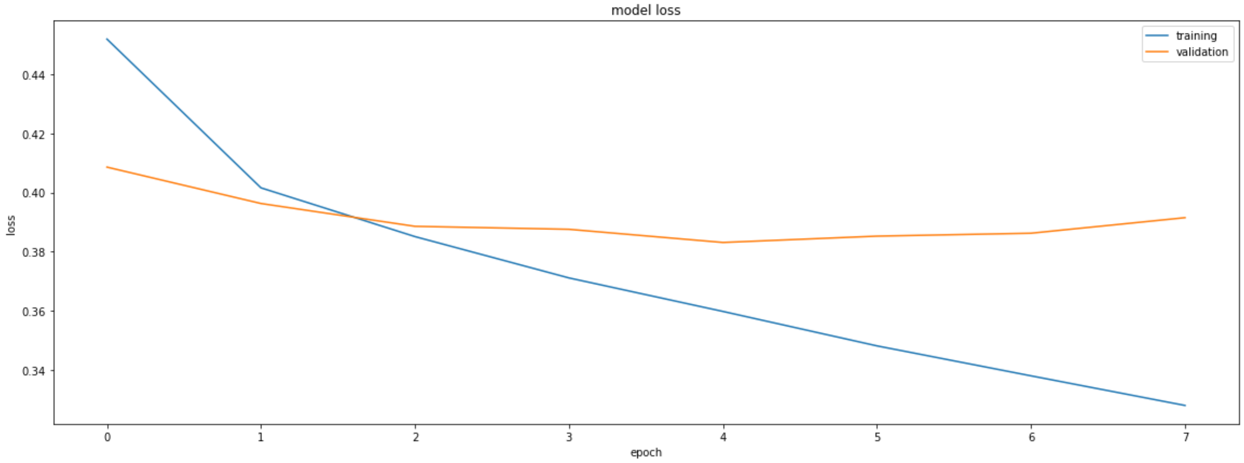
output sequence Length	**output_dim**	**Test Accuracy**	**Test Loss**	**Recall**	**Precision**	**F1 Score**	**RMSE**
None	64	0.865	0.39	0.87	0.87	0.87	0.655
None	128	0.864	0.391	0.86	0.86	0.86	0.649
None	256	0.859	0.4	0.86	0.86	0.86	0.658
96	64	0.859	0.394	0.86	0.86	0.86	0.655
96	128	0.871	0.38	0.87	0.87	0.87	0.638
96	256	0.864	0.388	0.87	0.87	0.87	0.647

In experiments 1 to 6, we used a constant dropout regularization rate of 0.2. With optimal output sequence length 96 embedded to 128-dimension word vectors, let us experiment with dropout regularization hyperparameter tuning in experiments 7 and 8.

Experiment 7 – LSTM, Output Sequence Length 96, Embedding Dimension 128, No Dropout

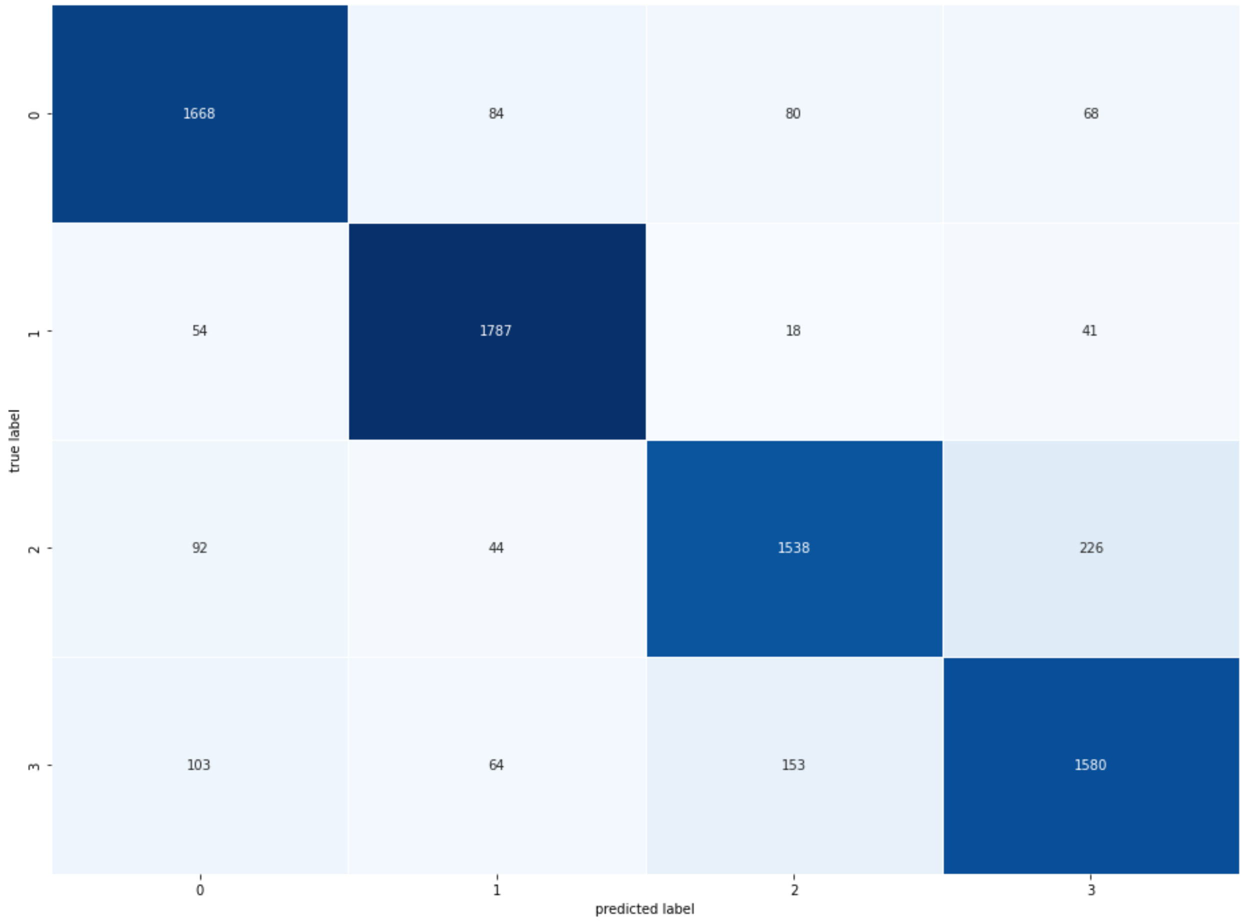
This experiment uses an LSTM model with an output sequence length of 96 and embeds the articles to 128-dimension word vectors. Dropout regularization is not used in this model. We chart the training and validation accuracy and cross entropy loss below.





From the charts above, we see validation results start to diverge from the training results after 4 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 86.3%, 0.8% worse compared to our most optimal model so far.

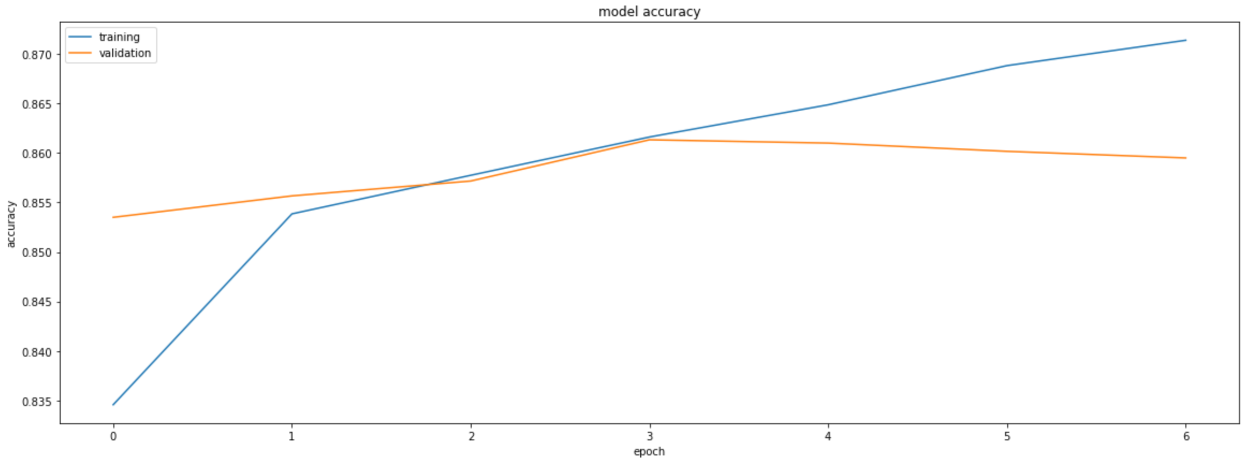
The experiment yields the following confusion matrix:

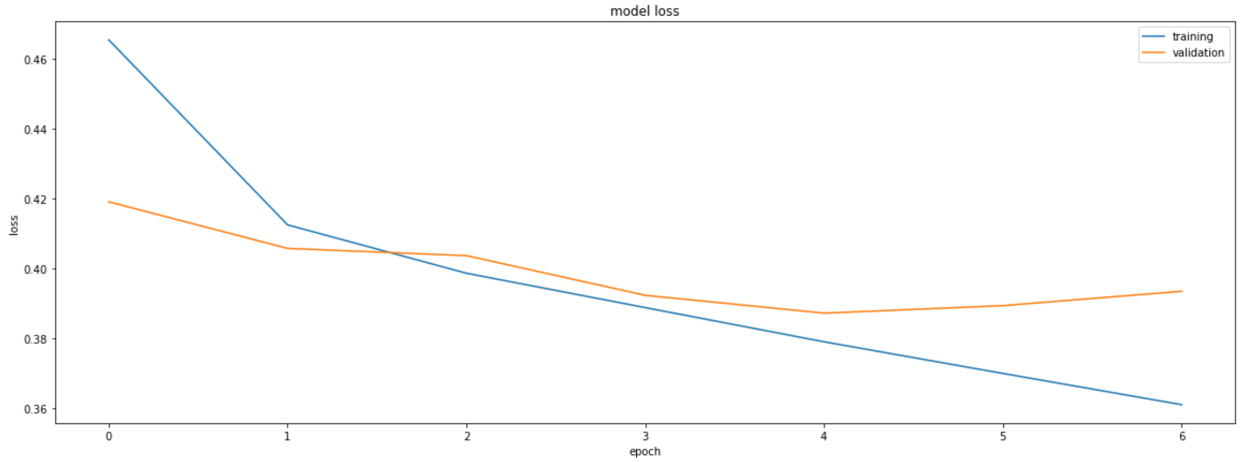


As demonstrated in previous models, the confusion matrix above shows this model has some difficulty correctly categorizing articles between business and science/technology.

Experiment 8 – LSTM, Output Sequence Length 96, Embedding Dimension 128, Dropout 0.3

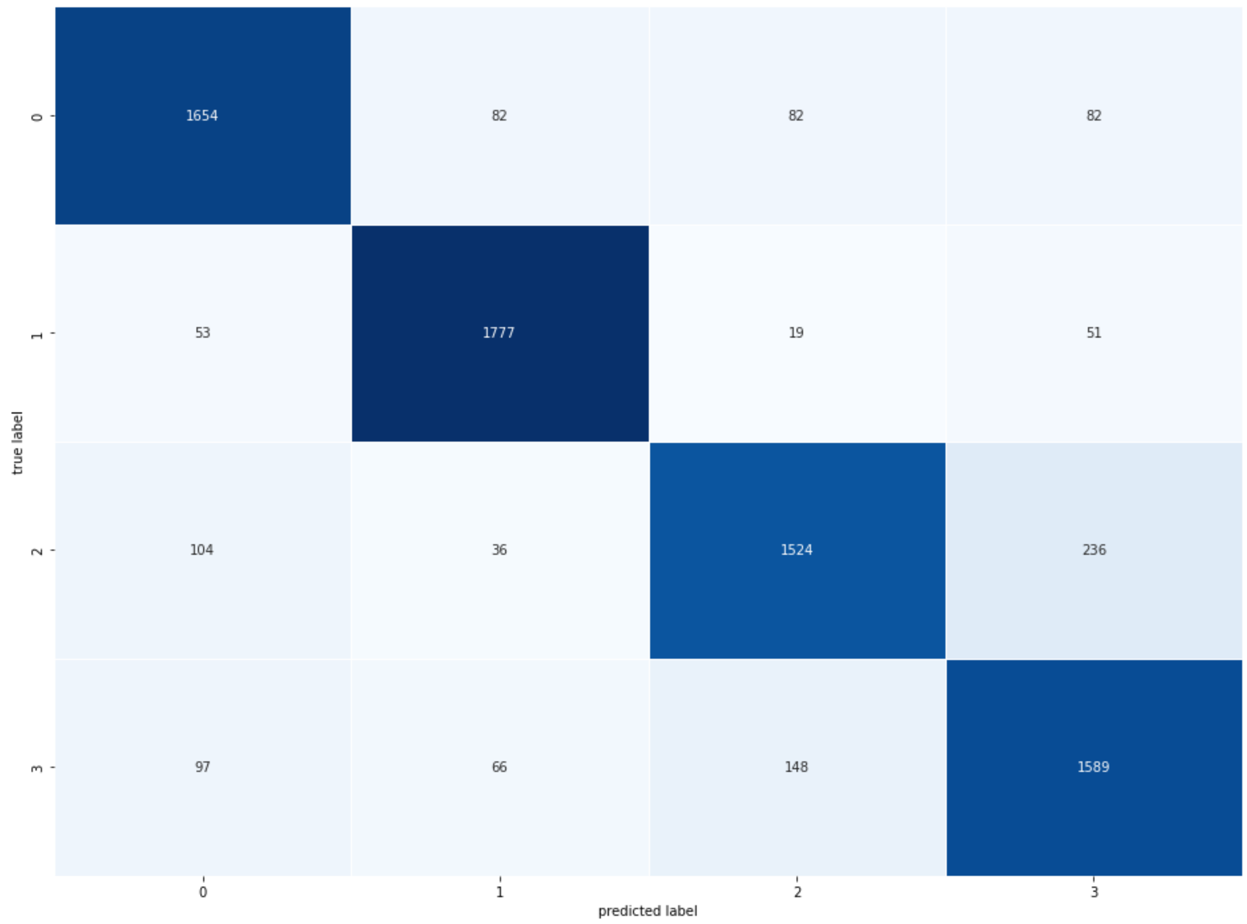
Experiment 8 is an LSTM model with an output sequence length of 96 and embeds the padded articles to 128-dimension word vectors. The dropout regularization rate is set to 0.3. We chart the training and validation accuracy and cross entropy loss below.





Validation results start to diverge from the training results after just 2 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 86.6%, better than experiment 7 with no dropout but still worse than our most optimal model so far with a 0.2 dropout rate.

The experiment yields the following confusion matrix:



As we can see from the confusion matrix, changing the dropout rate does not improve the mis-categorization between business and science/technology articles.

A table comparing LSTM models based on dropout rate regularization is below. We found the model using a rate of 20% yielded the best test accuracy on the AG News test dataset with 87.1%.

LSTM - Dropout Regularization

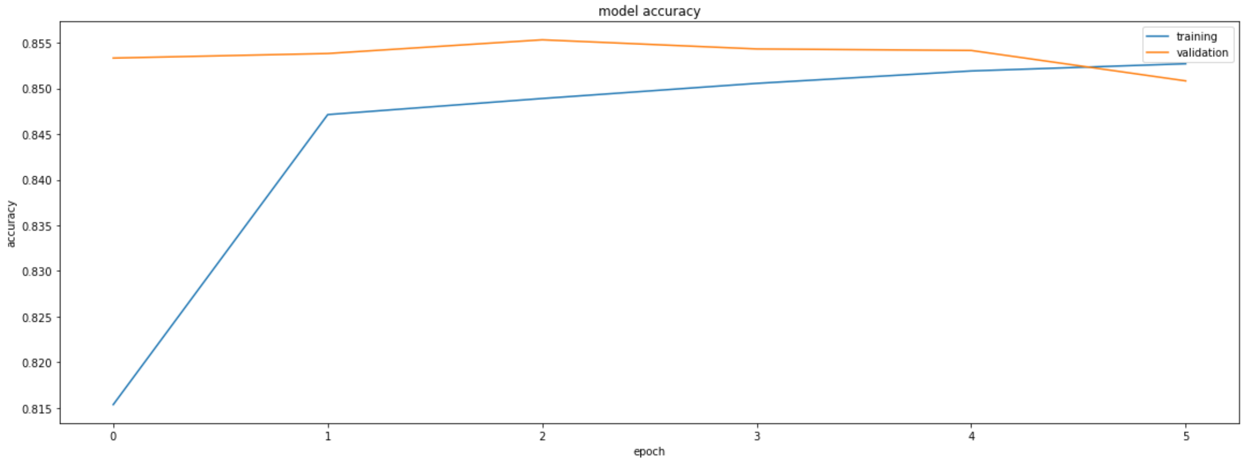
Output Sequence Length: 96, Embedding Dimension: 128

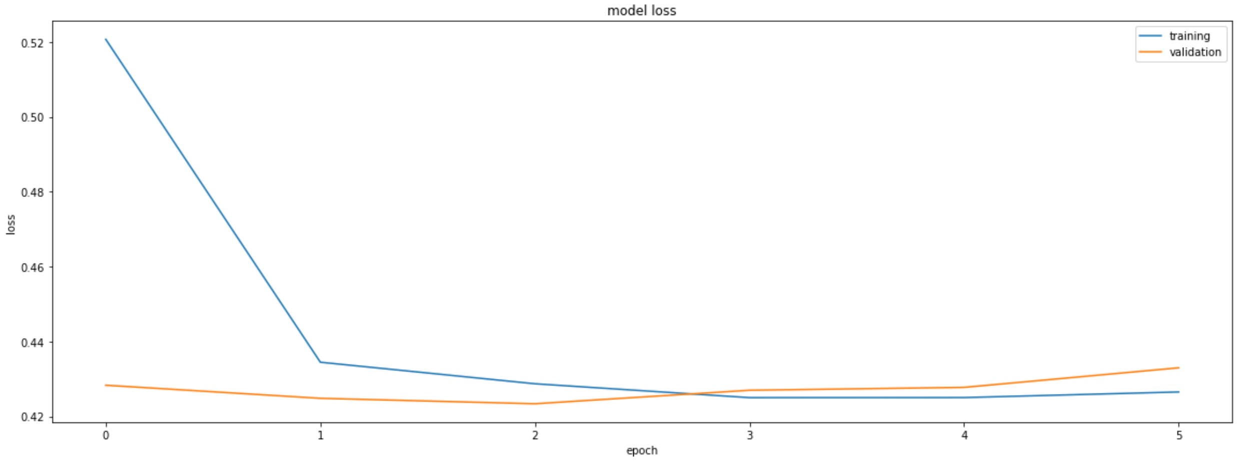
Dropout rate	**Test Accuracy**	**Test Loss**	**Recall**	**Precision**	**F1 Score**	**RMSE**
None	0.863	0.392	0.86	0.86	0.86	0.652
0.2 (starting value)	0.871	0.38	0.87	0.87	0.87	0.638
0.3	0.859	0.399	0.86	0.86	0.86	0.669

The following experiments below will be based on developing 1D convolution models.

Experiment 9 – 1DConv, Output Sequence Length 96, Kernel Size 3, 32 Filters, Dropout 0.2

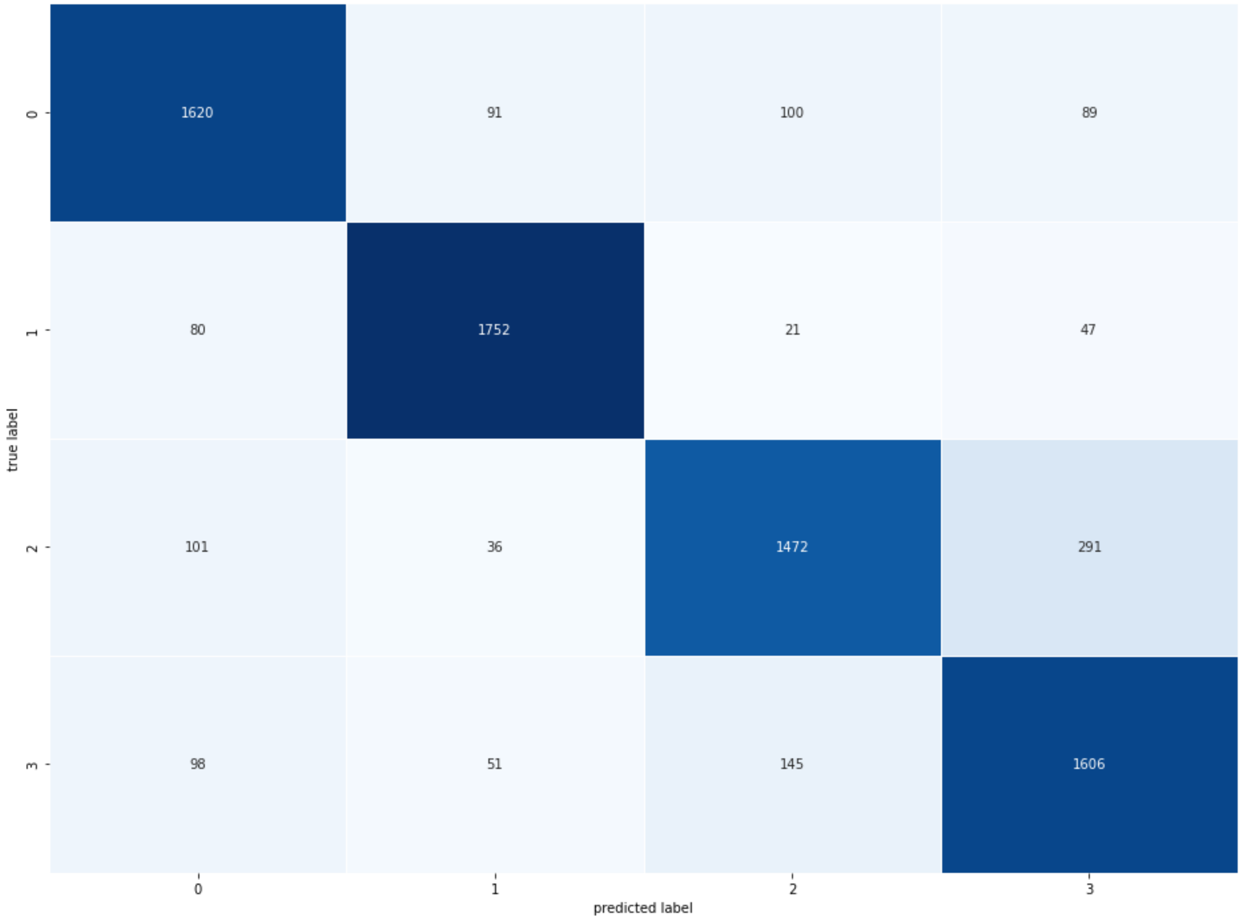
Starting with experiment 9, we now switch to 1D convolution models. This model uses an output sequence length of 96 and is embedded to word vectors via one-hot encoding. The convolution layer is configured to kernel size 3 and 32 filters. The dropout regularization rate is set to 0.2. We chart the training and validation accuracy and cross entropy loss below.





Validation results take longer to diverge from the training results at about 5 epochs, indicating signs of the model overfitting to the training data. The resulting test data accuracy was 84.9%, worse than any LSTM-based model. Let's see if increasing the kernel size in the next several experiments improve model accuracy.

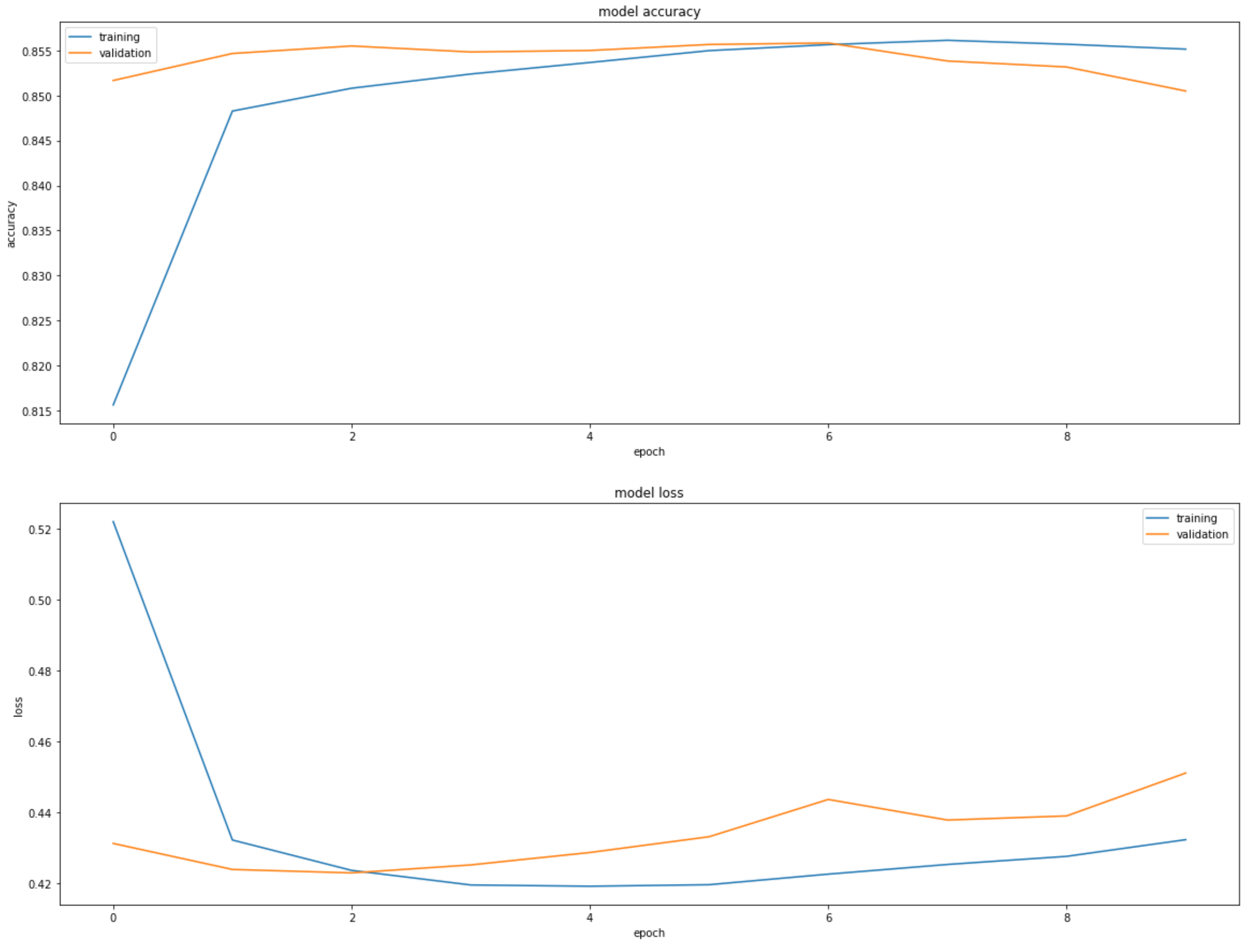
The experiment yields the following confusion matrix:



Despite changing to a 1D convolution architecture, the confusion matrix above shows this model also has some difficulty correctly categorizing articles between business and science/technology.

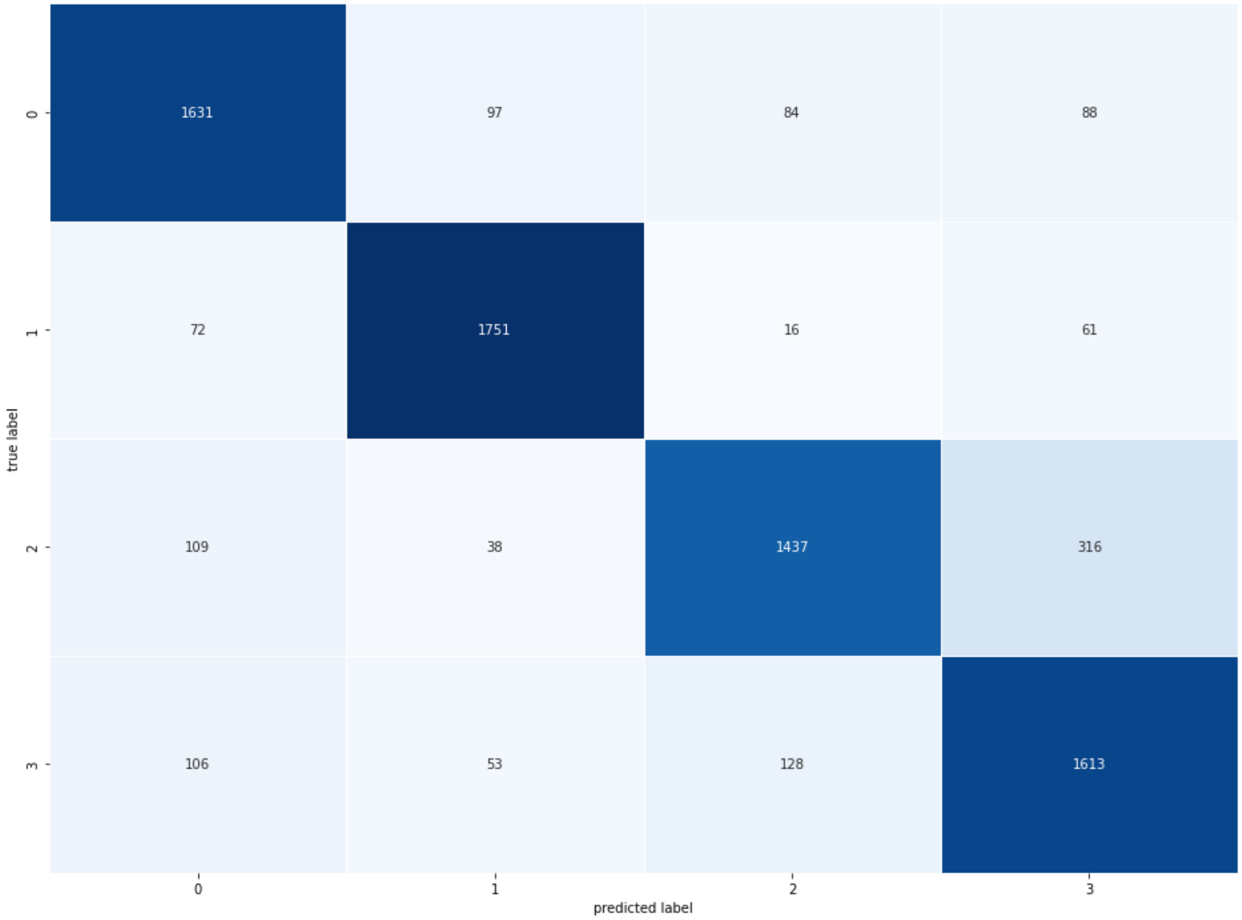
Experiment 10 – 1DConv, Output Sequence Length 96, Kernel Size 4, 32 Filters, Dropout 0.2

Experiment 10 is a 1D convolution model using an output sequence length of 96 with the articles embedded to word vectors via one-hot encoding. The convolution layer is configured to kernel size 4 and 32 filters. The dropout regularization rate is set to 0.2. We chart the training and validation accuracy and cross entropy loss below.



Validation results start to diverge from the training results at about 6 epochs. The resulting test data accuracy was 84.6%, our worse model so far. Increasing the kernels even more might lead to worse results, but we will continue to experiment and observe the results.

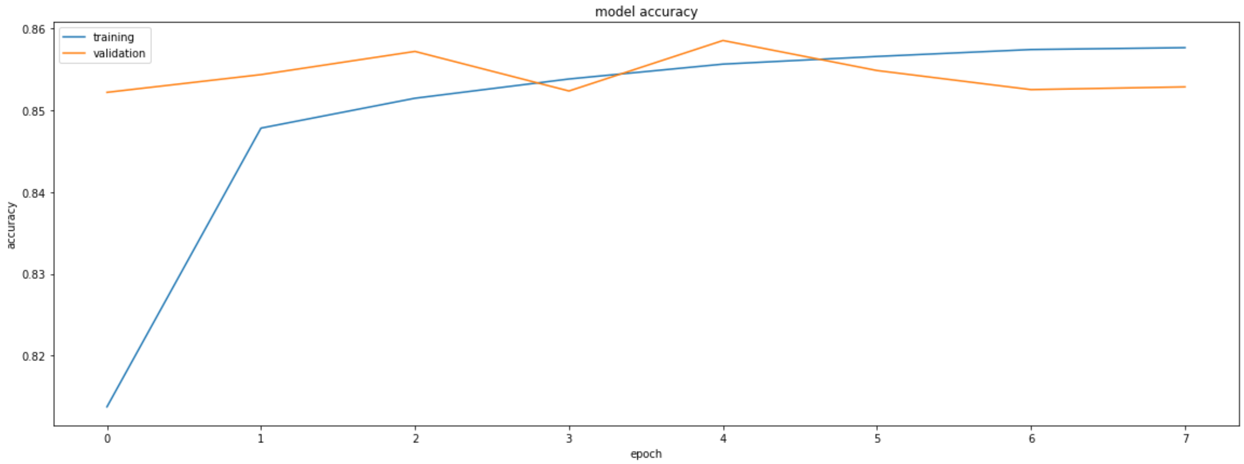
The experiment yields the following confusion matrix:

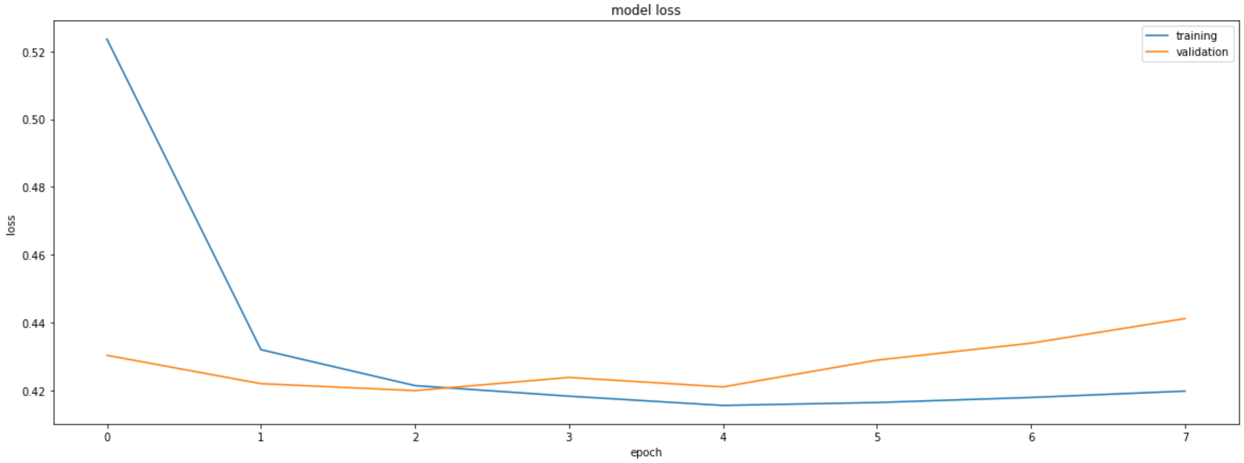


The confusion matrix shows some mis-categorization between business and science/technology still exists and is not improved.

Experiment 11 – 1DConv, Output Sequence Length 96, Kernel Size 5, 32 Filters, Dropout 0.2

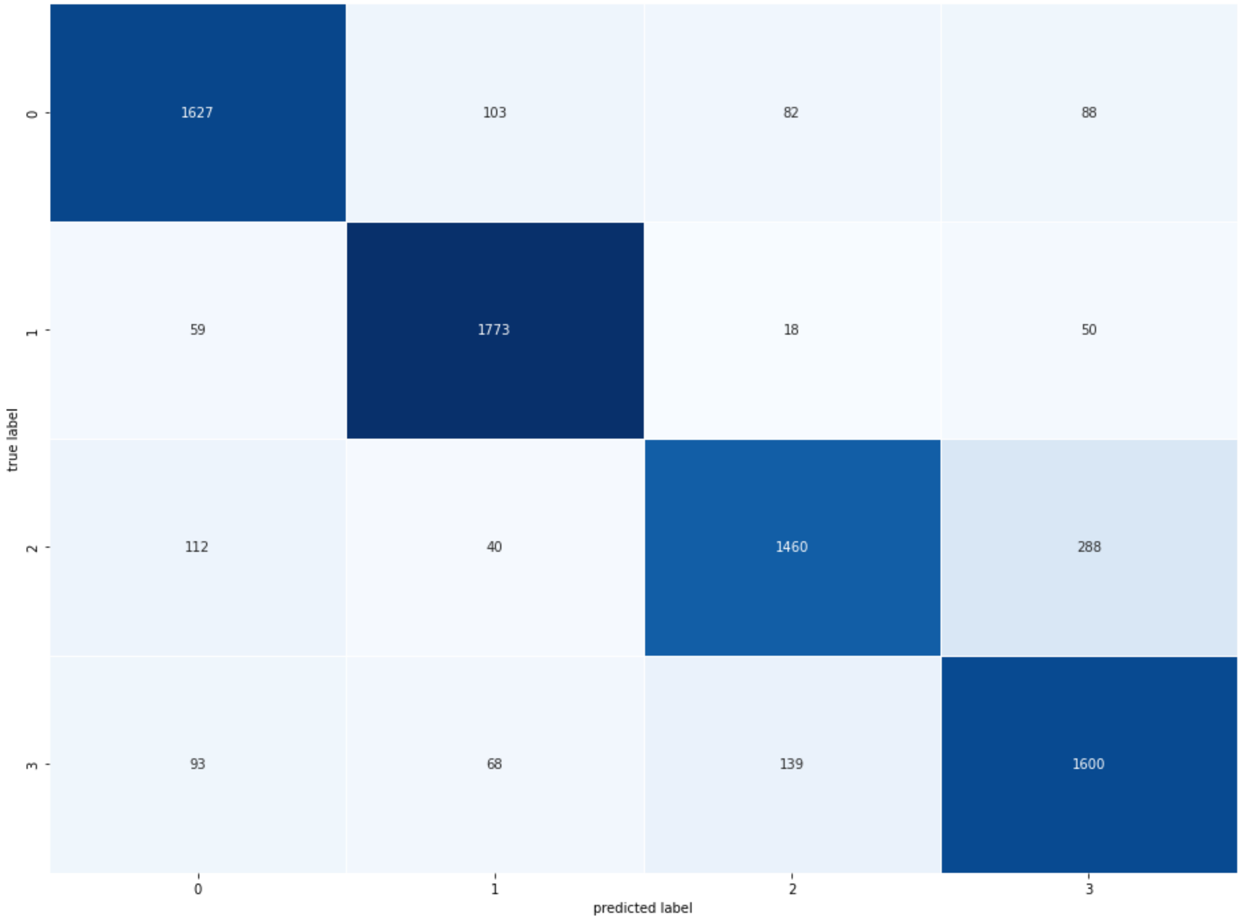
This is a 1D convolution model with an output sequence length of 96. Like all of our 1D convolution models, the articles are embedded to word vectors via one-hot encoding. The convolution layer is configured to kernel size 5 and 32 filters. The dropout regularization rate is set to 0.2. We chart the training and validation accuracy and cross entropy loss below.





Validation results start to diverge from the training results at about 4 epochs. The resulting test data accuracy was 85%, our best 1D convolution model so far, but still worse than any LSTM model we've made. We will continue to increase the kernels one more time.

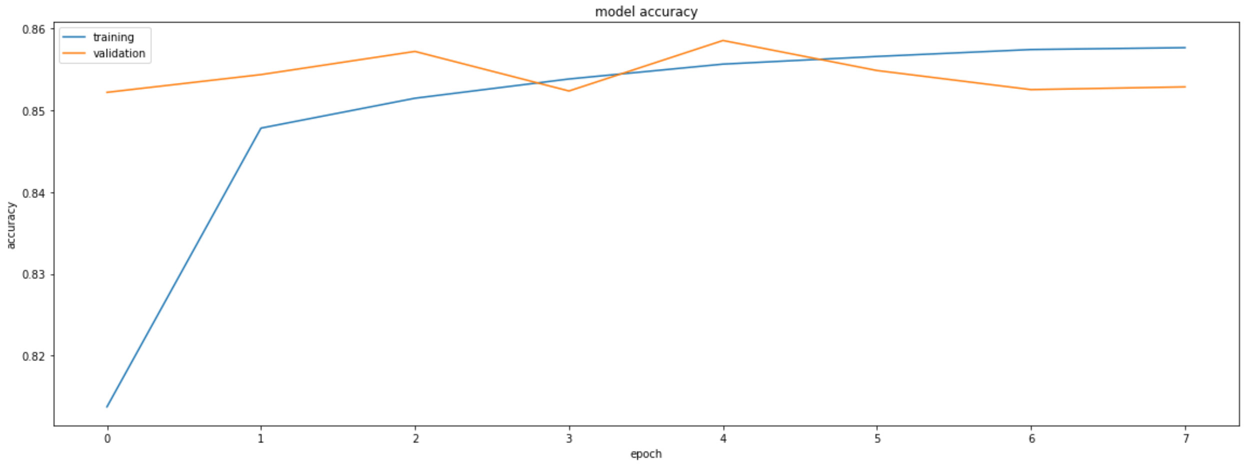
The experiment yields the following confusion matrix:

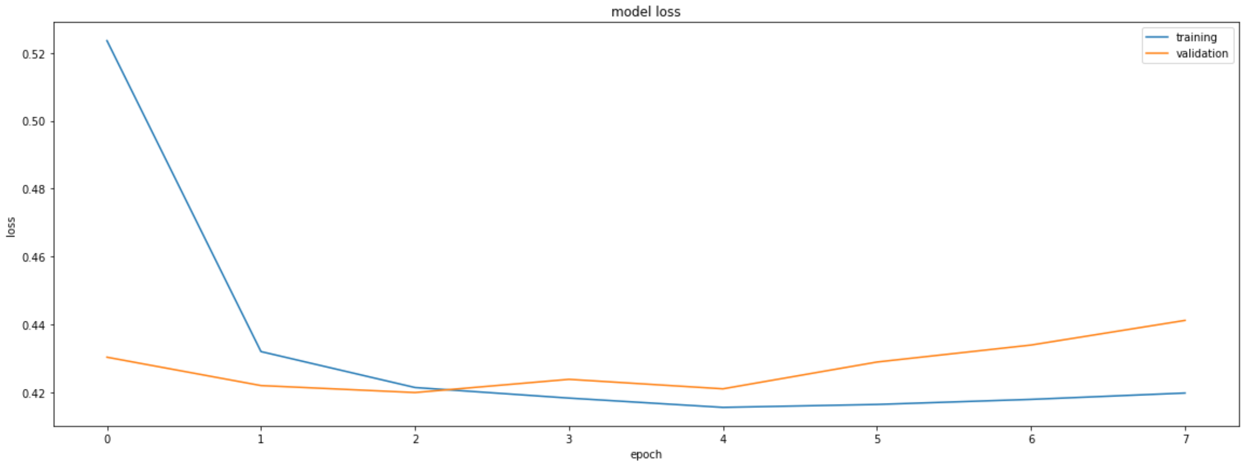


The confusion matrix shows some mis-categorization between business and science/technology still exists and is not improved.

Experiment 12 – 1DConv, Output Sequence Length 96, Kernel Size 6, 32 Filters, Dropout 0.2

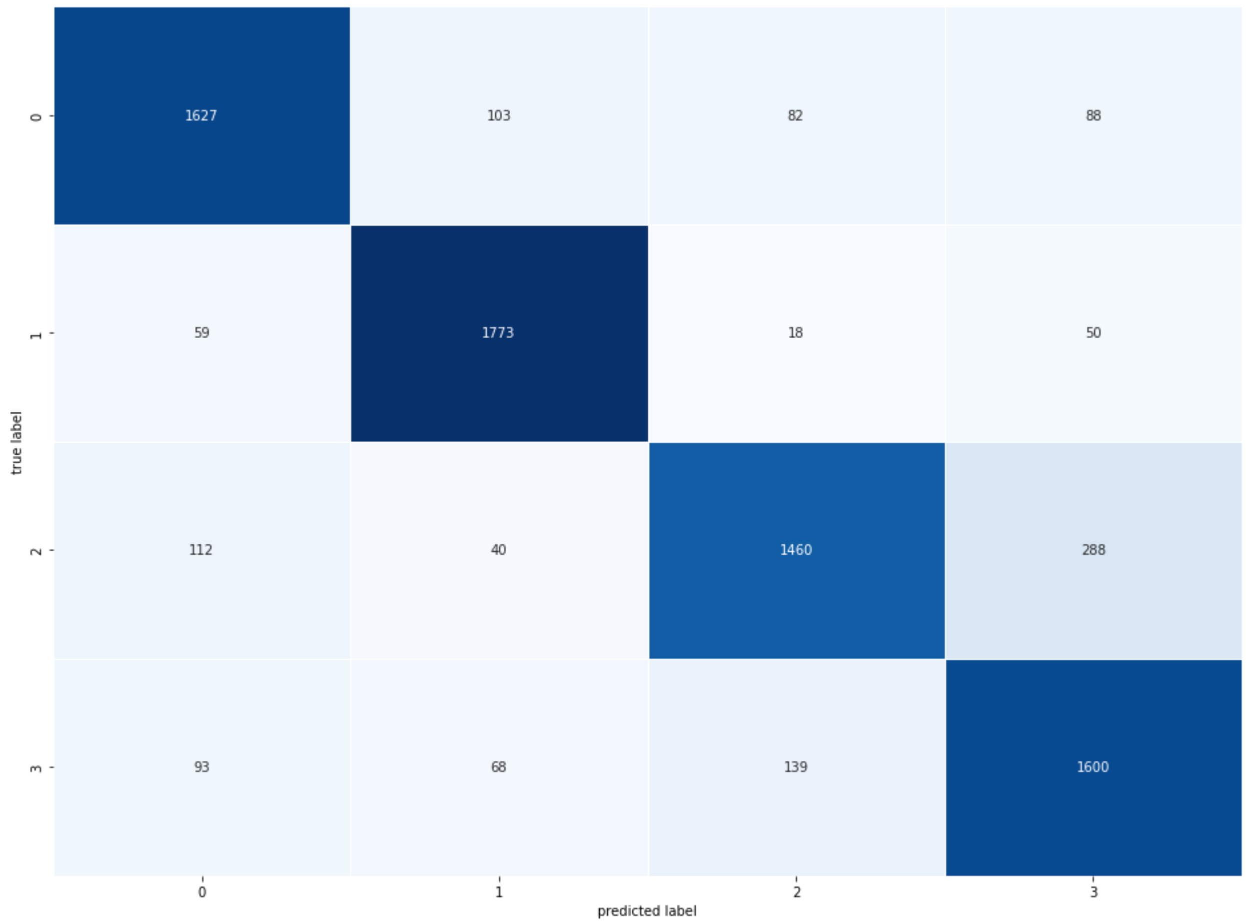
Experiment 12 is a 1D convolution model similar to that in experiment 11 but with the convolutional layer configured to kernel size 6. We chart the training and validation accuracy and cross entropy loss below.





Validation results start to diverge from the training results at about 4 epochs. The resulting test data accuracy was 84.7%, which is not as good as in experiment 11.

The experiment yields the following confusion matrix:



Some mis-categorization between business and science/technology articles still exists in this model.

A table comparing our 1D convolution models based on kernel size is below, in which we find the models using kernel size 5 yields the best test accuracy on the AG News test dataset with 85%, which is not as good as our best LSTM-based model.

1D Conv: Kernel Size

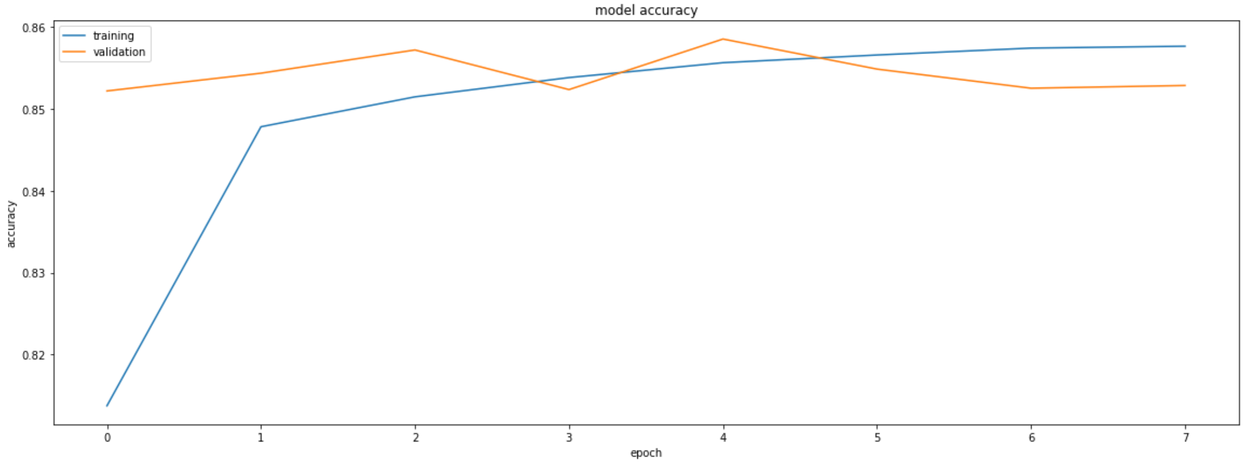
Dropout 0.2, Filters 32

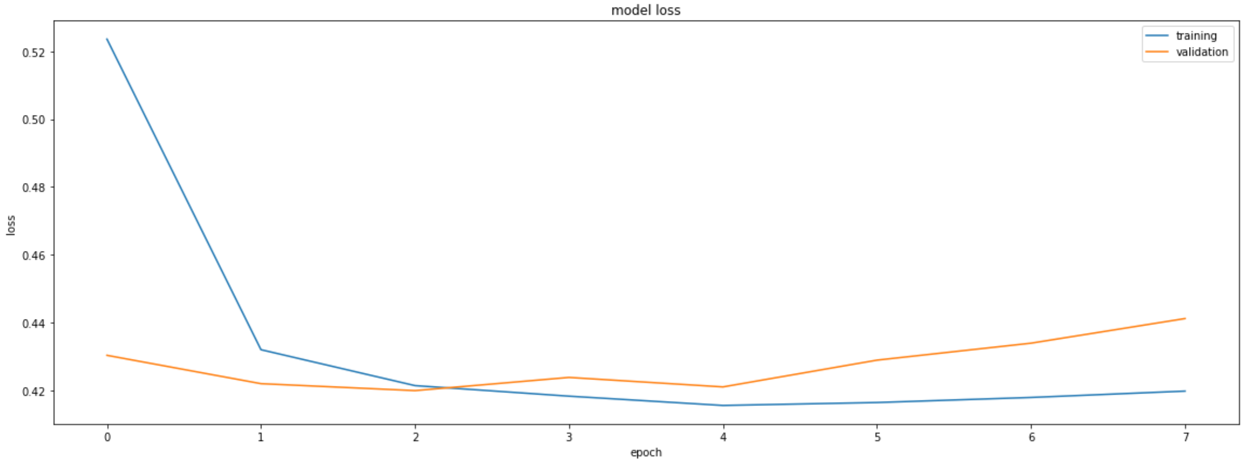
output sequence Length	kernel_size	Test Accuracy	Test Loss	Recall	Precision	F1 Score	RMSE
96	3 (Start)	0.849	0.437	0.85	0.85	0.85	0.683
96	4	0.846	0.439	0.85	0.85	0.85	0.692
96	5	0.85	0.435	0.85	0.85	0.85	0.681
96	6	0.847	0.436	0.85	0.85	0.85	0.681

We found that the 1D convolutional model using kernel size 5 yielded the best accuracy on the AG News test dataset with 85%. With this kernel size constant, we will experiment with increasing the filter size next.

Experiment 13 – 1DConv, Output Sequence Length 96, Kernel Size 5, 64 Filters, Dropout 20%

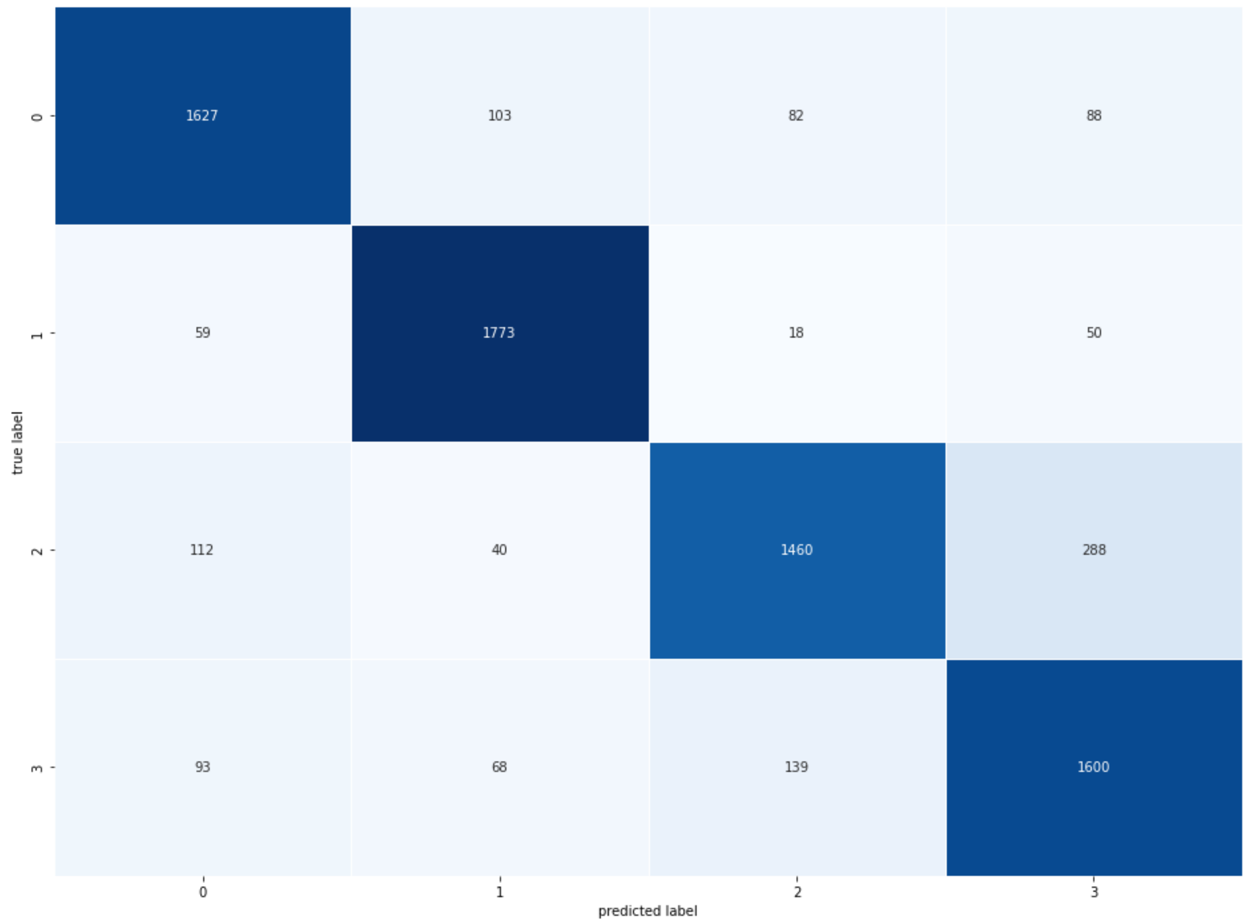
Using optimal kernel size 5, this experiment is a 1D convolution model similar to that in experiment 11 but configured to 64 filters in the convolution layer. We chart the training and validation accuracy and cross entropy loss below.





Validation results start to diverge from the training results at about 4 epochs. The resulting test data accuracy was 85%, which is the same as out best 1D convolution model so far.

The experiment yields the following confusion matrix:



Some mis-categorization between business and science/technology is a recurring theme that exists in all models we have developed.

A table comparing 1D convolution models based on filter size is below. While the model with 64 filters performs the same as the model with 32 filters in terms of test accuracy, the test loss is improved by 1.3%.

1D Conv: Filters Size

Dropout 0.2, output sequence Length 96, Kernel Size 5

Filters	kernel_size	Test Accuracy	Test Loss	Recall	Precision	F1 Score	RMSE
32 (Start)	5	0.85	0.435	0.85	0.85	0.85	0.681
64	5	0.85	0.422	0.85	0.85	0.85	0.681

The best 1D convolutional model we found yielded 85% accuracy based on tuning dropout and kernel size hyperparameters, our best overall model was an LSTM-based model yielding 87.1% on the AG News test dataset.

Conclusion

From the experiment results above, our best text classification model was able to achieve an 87.1% accuracy on the AG News test dataset. If this accuracy exceeds the given threshold requirements, we should go into the next phase of converting the LSTM-based model framework along with its hyperparameter configurations and training weights to a pre-trained model which will then be integrated into a full model to be trained on a dataset of our historical customer service chat logs. That being said, while the infrastructure of capturing historical chat logs is in place, we will need to have at least a year's worth of those chat records extracted, curated, and categorized to create that dataset. We will need to develop a data engineering process to clean the logs into sets of customer chat messages as well as create a set of customer service categories that covers the most common customer chat situations. We will also need people who can spend time on the task of manually categorizing chat logs. Depending on the volume of data, we can consider outsourcing that effort to Amazon Mechanical Turk.

While we are able to develop a text classification model that exceeds the performance requirements on the AG News dataset, there is still more work to be done before it can generalize to our customer service chat logs dataset and be fully integrated to our chatbot system.

Resources

Cheng, J., Dong, L., & Lapata, M. (2016). *Long Short-Term Memory-Networks for Machine Reading*. arXiv.org. https://doi.org/10.48550/arxiv.1601.06733

Kim, Y. (2014). *Convolutional Neural Networks for Sentence Classification*. arXiv.org. https://doi.org/10.48550/arxiv.1408.5882