Mid Term Report

Magdaleno, Alejandro
amagale@student.kennesaw.edu

Lyles-Woods, Quin'darius
qlyleswo@student.kennesaw.edu

October 17, 2021

Progress

The goal of this research is to build a classification model to determine where bodies of text originate from using currently a Sequential Model from the Keras Library. The progress that we have had so far is multifold. For the data, we have been able to write a script that allows us to get 40-60 articles at a time and strip them of only their stories and outputting that data into a folder properly labeled with the classes that we need. So the current article count is around 120 and the token count is around $124,998$ and will grow at the pace that new articles are produced. We have a pipeline for creating a vocabulary and frequency count for the current data set. With this pipeline that directly helps us create vectors our data and then use that information for our sequential model. Our loss values with this model are 20% and the Accuracy that the model is achieving now with this limited data is 97%.

Methods

There were a few considerations when approaching this project. Two of the main ones were using pipelines provided by fasttext, spacy, and TensorFlow. Fastttext provided a very quick and easy route to create a supervised classification model for our text with its pipeline. The one problem with it however was trying to be flexible and account for complexity within the text. Due to this, TensorFlow ended up being the main library used in our text classification project. With the use of TensorFlow, we were able to label our data through the use of directories. This means that TensorFlow can detect class "a" and class "b" within the test and train directory folders of our data. As we get more data, TensorFlow is flexible enough to adjust to the increase in classes when we had more news stations to our training and test folders to add more classes later. Currently, the model takes in the raw text files that we put together and vectorizes the text into integer vectors. The model will create a vocabulary and frequency with these vectors in order to process the text embedding. After we setting up the input into integer vectors, a neural network is created through TensorFlow Keras. Currently, a sequential or feed-forward network is being used as the base model. This model can be adjusted if needed into other sorts of models like convolutional networks or recurrent networks. A neural network ended up being chosen here to provide flexibility and complex learning capabilities to the problem. Our neural network uses the adam algorithm as the optimizer which is a stochastic gradient descent algorithm provided by Keras. The loss function used is binary cross-entropy which is only being used while we have two different classes. This model still has plenty of adjustments being made as more data comes in. Such as changing hyper-parameters, learning rates, loss functions, optimizers, and the architecture of the neural network.

Results

The base model has decent results but this could be due to the amount of data being used. After training using 80 percent of all data, we were able to get an accuracy of 97 percent and a loss of 20 percent. Currently, these are the main numbers that we are focused on and are trying to adjust. 97 percent for the accuracy is a great number but we want to make sure the model didn't overfit the data. This would be a problem that we could fix by adding in more data so that the model has more to see. If the model starts learning really well from little data, it will start to predict the next results almost as if it's remembering previous results. We want to avoid that problem down the road. By continuing that process of adding data and increasing the complexity of the model, we will also be able to bring down the loss further and even avoid underfitting. Underfitting would be where the model can't make good predictions of the data. Currently, this seems to be avoided which is good and allows us to focus on other aspects of the project. Work to predict on new test sets is still being made. Working with tensorflow datasets is new and we are looking into creating new test sets with these data types in order to give the model new articles to predict.

Challenges

So far the biggest challenge has been the dataset. Collection and cleansing of the data has been a difficult thing to do. The main issue is that there were no pre-existing data set that had the data in the format that we needed to build such a classification model. So we are currently building our own which allows for greater flexibility when building the models but the start up cost are significant. The process isn't ironed out completely but its working currently and is a good base to start from. We still need some more labels when pulling the data to reach some of the other auxiliary goals that were mentioned in the proposal. We are helped with the Tensor Flow library automatically labeling the data from the directories. Getting the title of the article and assigning it to the file name would allow us to gather more information from our data set. Creating a working model to our standards has not posed a great challenge, currently we are getting 97% with very limited data but as said in the methods these results could be gilded with our small data set. Going forward we can see the finish line and we will polish the processes if we have more time left over.