

|  |
| --- |
| Deep Learning  CA1 |
|  |
| March 18  DBS  Authored by: Robert Clark  Student Number: 10596311  Student Email: 10596311@mydbs.ie |

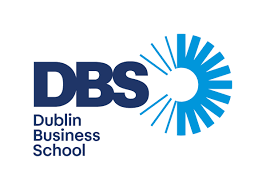


Table of Contents

[Text Classifier on Reddit Political Dataset 3](#_Toc99092236)

[Business Understanding 3](#_Toc99092237)

[Data Understanding 4](#_Toc99092238)

[Data Preparation 5](#_Toc99092239)

[Modelling 6](#_Toc99092240)

[ANN 7](#_Toc99092241)

[Model Background: 7](#_Toc99092242)

[Deep Learning Design: 7](#_Toc99092243)

[CNN 8](#_Toc99092244)

[Model Background: 8](#_Toc99092245)

[RNN (LSTM) 10](#_Toc99092246)

[Model Background: 10](#_Toc99092247)

[Deep Learning Design: 10](#_Toc99092248)

[Evaluation 11](#_Toc99092249)

[Deployment 12](#_Toc99092250)

[Bibliography 13](#_Toc99092251)

|  |
| --- |
| Text Classifier on Reddit Political DatasetBusiness Understanding In the partisan landscape of the United States of America many individuals are voting not just during elections but voting with their wallets. This behavior is not limited to America as many American practices are then propagated around world through American media. Businesses that take a political stance stand to both gain and lose customers and learning which way an individual’s politics lean can be greatly beneficial in understanding your marketplace and influence how you approach your advertising and corporate decision making. Using a model to determine political sentiment can be used to better understand your marketplace and trends that are occurring within it.  Added to this as was shown unethically by Cambridge Analytica, you can determine which individuals are likely to vote and in what direction they will vote, depending on the confidence and reach of your data you could predict an election, or at least better determine the most likely candidate. Different presidents will have very different impacts on the business landscape which can determining strategy and tactics for a four-year period in America. Having an indication of which candidate is more likely to win can be greatly enhance fiscal planning.  Thus, creating a deep learning model that can accurately predict an individual’s political stance from the title of reddit post, and then further training the model on other short form datasets like Facebook or twitter posts, then applying those insights can assist and inform decision making at the highest levels of a corporate structure. Data Understanding In creating this model, a dataset was needed. For this project we procured the Liberal vs Conservative dataset from Kaggle which is a dataset comprised of roughly 12000 reddit posts, and contains columns Title, Political Lean, Score, ID, Subreddit, URL, Number of Comments, Text, and Date created. A snippet of which is shown below.  Graphical user interface, application  Description automatically generated  For our purposes it was decided that the only columns to be used for this were title and political lean as we were creating a binary classifier model.  Thus the other columns were dropped in favour of focusing on these 2. Score, number of comments and Text are all valuable columns in other scenarios but were beyond the scope of this project for various reasons. Score and number of comments show engagement with an issue and in a scenario where we were using the reddit API to see how positive and negative these comments were we could better determine how divisive or popular a particular issue is, but are not useful for determining the political stance of the poster. Text is the most egregious column to be dropped but many of the text columns were empty. Specifically 10426 out of 12854 entries in the dataset. While filling these was an option, due to computational constraints and the potential noise created it was deemed unnecessary as the titles were all present. URL, ID, and Date were all left out as we were focusing specifically on the text analysis in this project and Subreddit was removed as it was potentially too strong an indicator outside of the textual data that we were looking to explore.  The data gathered was imbalanced, as shown in the picture below. Being skewed with roughly 65% of the posts being liberal leaning. This imbalance would need to be addressed in the data preparation stage of the project.  Table  Description automatically generated with medium confidence Data Preparation To prepare the data it was loading into Collab via the csv library and turned into a Pandas data frame from which it could be adjusted to be more useable. Firstly, we ensured that only the two relevant columns were present in the data frame. Then we adjusted the Title column to better suit our needs. Removing punctuation, and in certain circumstances converting it to lower case. This was functionalized to be used individually by each model on a case-by-case basis. Both the CNN and LSTM models were using tokenizers that lowered the text’s case themselves and thus this was a redundant step. For the ANN model we used a N-gram text vectorization to better capture certain sentiments. As certain figures are divisive the words that preceded and followed them can be very influential in the classification. For example, discussions on Donald Trump, could read, “I love Trump” or “I hate Trump”. Obviously having a very different likelihood of being a Conservative or Liberal based on this. Added to this we removed stop words for better classification, so instead of “I love Trump”, the sentence would be reduced to “Love Trump” or “Hate Trump”.  To return to N-gram’s and better explain their behavior, it involves breaking the sentences down into smaller chunks for tokenization, but instead of treating each word individually they treat each N words in a sentence as an individual token. Modelling Three models were chosen for exploration within the scope of this project, they were ANN, CNN, and RNN, specifically a LSTM variant. All three have different general areas they excel in, for example CNN networks are usually useful in computer vision tasks as they can treat images like matrices.  In terms of parameters for our models we use a binary cross entropy measure of loss as this is a binary classification problem. For our model optimizer we used adam optimization, which can be described as,  “Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models.”(Brownlee, 2017)  Instead of maintaining a single learning rate across all nodes for the duration of the training, the adam optimizer adapts the training weights of the based on the exponential moving average of the gradient, and the squared gradient, as well as parameters beta1 and beta2.  As this is a binary classification with little cost for a wrong classification, accuracy is the most important metric and is the one utilized in this model.  Generally, throughout the project we use sigmoid and relu activation functions. Rectified linear activation function is known as relu. This algorithm returns a value that is positive, or 0 if the value is negative. Relu is the most prominent activation function in neural networks currently as it acts like a linear function which allows better optimization. Relu helps reduce the issue of vanishing gradients, as the gradients remain proportional to the node activations. Sigmoid activation function is generally used in the final layer of the models, as it is optimal for binary classification problems. It outputs a number between 0 and 1 which is then converted into a classification based on being above or below .5. ANNModel Background: ANN stands for Artificial Neural Network, specifically in this case we are using a feed forward neural network. The data in these networks as the name suggests only travels in one direction, forward. In these networks there are multiple layers that interact, each neuron is fully linked to all other neurons in the adjacent layers, with each connection having a different weight that influences whether the node is “activated” or not, if the weights are above certain threshold’s the node is considered activated. This threshold is usually 0, and an activated node will generally be weighted as 1, with a deactivated node generally being -1. Feed forward ANN are some of the simplest neural networks and can be useful in dealing with noisy data. As can be the case with NLP matrices.  Diagram, schematic  Description automatically generated  Model: 1 ANN Deep Learning Design: For our ANN model, the most optimal solution was determined to be keeping the model simple. It consists of a single layer of 10 nodes, and an under-sampled dataset. This helps prevent any overfitting as the data is a sparse matrix which is noisy. Added to this we only use 10 epochs in fitting the model as after 10 epochs overfitting occurs which can be seen in this graph. The comparison of the below graph will be shown in the evaluation section of this report.  Graphical user interface  Description automatically generated with medium confidence  Figure 1: Feed Forward 100 epoch accuracy and loss graph CNNModel Background: CNN stands for Convolutional neural network, these models take input and assign weights and biases to various specific points in the data, they require little preprocessing in comparison to other models, as they determine points of interest themselves. The input in a CNN model is 4-dimensional array, in an image it would consist of width, height, depth, and batch size, which is consistent with NLP. The Height and width are the size of the matrix created in advance of running the model, and the depth is the value of each cell of the matrix. Batch size being self explanatory. Convolutional layers are the layers of a model where filters are applied to the feature matrix input, reducing it to a new convolved feature map reduced in size. They apply this by moving through the matrix, multiplying the weights but the features, and the bias to produce a convolved feature, as shown in the below diagram.  A picture containing diagram  Description automatically generated  To achieve a matrix of the correct size in an NLP space we use padding to ensure that all rows have the same width by adding 0 values in pre or post manner. The matrix will be flattened at a later stage in the model. CNN models are less useful for NLP data that isn’t very generalized however they can produce near state-of-the-art results on large very generalized datasets.  Deep Learning Design:  In our model specifically we have constructed it using dense layers with relu activation function, a flatten layer to change the data from its former format to get a multiplication of the row and column of the previous matrix. This layer comes at the end of the model only followed by a single neuron sigmoid layer for final output. There is also a dropout layer which is useful in preventing overfitting. We use a rate of .5, multiple options were attempted but the best results were achieved with this dropout rate. Diagram  Description automatically generated RNN (LSTM)Model Background: RNN stands for recurrent neural network. They are known as recurrent neural networks as information within this model is passed backwards and forwards. They were initially derived from feed forward networks but with the implementation of back propagation RNN’s were born. They are most suited to series-based tasks like stock market prediction. LSTM stands for Long Short-Term memory. This model has 3 gates within it, input, forget and output. Simply explained the function of the input gate is to determine whether to let new information in or not, the forget gate is to determine if information isn’t useful and should be forgotten and the output gate is to propagate the data forward and affect the overall model. Using LSTM’s, we can reduce the chance of vanishing gradient problems as the gradients are steep enough meaning training is short and accuracy is high generally. Data input for LSTM models is 3 dimensional, with the first dimension representing batch size, the second dimension representing time step and the third dimension being input units. This means Deep Learning Design: Our RNN model is comprised of seven layers, an embedding layer being the first, embedding layers are used to convert the sparse matrix of a tokenized bag of words into a dense vector representation. It is comprised of an input dimension, which is the size of the vocabulary to be used in the data, the output dimension which is the size of the vector space the words will be condensed into, and the input length, which is the length of the documents your using, or the maximum allowed value. If your longest title is 1000 words, then that is the maximum length. In our data exploration prep stage, we discovered our longest title name is 300 and that is used here. Evaluation To evaluate the networks properly we must take each independently. We achieved a 74% accuracy rate in our ANN model, as shown in the below graph.  Chart, line chart  Description automatically generated  From the graph we can see that as the model continues it begins overfitting, which is why we chose a lower epoch number, initial epochs of 30 and 10 resulted in accuracy hovering around 60% and 70.1%. The ANN model in our test is the one that best cuts out the unimportant data and predicts the output correctly. In our prediction of novel title titles pulled from more recent reddit posts in partisan subreddits was correct on all three predictions.  Graphical user interface, text, application  Description automatically generated  However, the CNN and LSTM were less successful, while the LSTM achieves accuracy of 72% in the final test it is unsuccessful at its final prediction erroneously predicting Liberal political leanings in all cases. CNN was worse achieving accuracy of 64%. This is likely in part to do with the embedding. While the self-learning aspects of a CNN make it ideal for images, due to the lack of options for text vectorization our output was less able to determine valuable information from the noise and suffered as a result. Deployment As the models have been created and pickled at the end it could be deployed to a website to be used in predicting political leanings of reddit users. Were this model to be connected to an API pulling data from the reddit website on a continual basis it could be used in conjunction with some of the functions for data cleaning to aggregate the data and determine how partisan each subreddit is based on number of clearly partisan posts each political subreddit produces. Added to which with a little adaption this model could be applied to a wider variety of data sets i.e., tweet classification.  Notebook used can be accessed via: https://colab.research.google.com/drive/162tHAHEd0XwkfgQTXs83lTjb7\_A4dnEL?usp=sharing |

# Bibliography

Brownlee, J., 2017. *machinelearningmastery.com.* [Online]   
Available at: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/#:~:text=Adam%20is%20a%20replacement%20optimization,sparse%20gradients%20on%20noisy%20problems.  
[Accessed 18th March 2022].

Donges, N., 2022. *builtin.com.* [Online]   
Available at: https://builtin.com/data-science/recurrent-neural-networks-and-lstm  
[Accessed 3 March 2022].

Newatia, R., 2019 . *medium.com.* [Online]   
Available at: https://medium.com/saarthi-ai/sentence-classification-using-convolutional-neural-networks-ddad72c7048c  
[Accessed 8 March 2022].

Sharma, P., 2020 . *machinelearningknowledge.ai.* [Online]   
Available at: https://machinelearningknowledge.ai/different-types-of-keras-layers-explained-for-beginners/  
[Accessed 20 Feburary 2022].

Stewart, M., 2019 . *towardsdatascience.com.* [Online]   
Available at: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac#:~:text=Convolutional%20layers%20are%20the%20layers,the%20size%20of%20the%20kernels.  
[Accessed 8 March 2022].

Tensor Flow, 2022. *www.tensorflow.org.* [Online]   
Available at: https://www.tensorflow.org/text/tutorials/text\_classification\_rnn  
[Accessed 13 March 2022].