Binary Sentiment Classification

Prerna Tripathi

June 8, 2025

1. EDA on the dataset

The first thing to do was to check the shape of the train and test file to find out the size of the dataset been given to us. The next step is to find whether there are rows with missing values such that they can be dealt in the preprocessing stage. Third is to find the number of unique entries which can give us a more clear data about the actual useful part of the dataset. Next and one of the most important parts is to check whether or not is there a class imbalance by plotting both the polarity labels. On doing the EDA there was no class imbalance and almost similar number of training examples for both the classes. The final step was to check the distribution of lengths of the text in the dataset.

2. Stratified Sampling

The results of the EDA showed that our dataset was too large and hence a stratified sampling was performed on the train and test files in such a way that no class imbalance is introduced and the review length distribution remains almost similar. The desired sample size for train file is 50K and the test file is set at 10K. The sampled data was further checked to ensure there was no class imbalance introduced and the review length distribution did not change much. From the review length distribution obtained we also find out the desired padding length.

3. Data Preprocessing

After sampling the datasets in the desired size text preprocessing steps were performed. To begin with first all the missing vallues in the title and text columns were filled with empty string and then the title and text were merged into a single column only for easier analysis of both text and title. Next step was to drop the title column. Then the polarity was mapped to a conventional 0-1 form where 0 is for negative and 1 for positive. Then the text

was handled for contractions and was made devoid of any punctuation marks, it was made lowercase and it was tokenized. Finally it was padded to a max sequence length of 150.

4. biLSTM model

In both the biLSTM and biRNN models the GloVe pretrained embeddings with a dimension of 300 has been used. The 300 dimension was chosen for better analysis and accuracy in case of sarcastic reviews. Also here the a two directional variant of LSTM and RNN both have been used so that the review text is analysed in both forward and backward directions to grasp the context of the statement well and improve the accuracy of the model. Next the lstm layer was defined with the input size equal to the embedding size of 300, the hidden state size was set at 128 and the sequence length size was set from the EDA analysis at 150. After the lstm layer the linear classification layer was defined with the input layer of size same as the hidden state size and then the output layer of size 2 since we have 2 classes to predict.The output is obtained by applying the softmax function on the outputs of the input layer. The loss function is the cross entropy loss and the learning algorithm for learning the parameters is the Adam,that is gradient descent with adaptive moements with the number of epochs as 10 and the learning rate at 0.001. Also to prevent overfitting a dropout of 0.5 has been set for the final hidden state which is the input for the classification layer. The training dataset was also split into a training and validation dataset to find the best generalising model and not the overfitted model.The batch size for the training and validation set was set at 128. Then finally the the model was tested over the test file with a batch size of 64 and the accuracy,f1 score and the confusion matrix was printed.
Accuracy= 0.9042
F1-Score= 0.9035
Confusion Matrix=[[4559 442][516 4484]]

5. biRNN model

The only thing different in the biRNN model is just in the defining of the model to be used and the class of the biRNN module. Rest all the other things are same as for the biLSTM model.
Accuracy= 0.8329
F1-Score= 0.8156
Confusion Matrix=[[4303 698][797 4203]]

6. Analysis of False positives and False negatives

6.1. Why might the model classify this and what clues can humans pick up?

Some of the false negatives include texts which have too much of negation words like 'not' due to which the model is not able to detect the positive meaning behind it. Humans on the other hand can easily figure out the meaning of the statement because of their high intelligence and understanding of texts. In some reviews the writer liked most of the things but disliked others causing multipolarity in the statement so the model falsely classified it as negative. Too casual tone in the reviews like 'a chick flick','lol' will cause difficulty for the model to rightly classify them. Some false positives reviews included sarcastic reviews of the writers. In one such review the writer said that 'her family loves the set' which had a sarcastic tone to it which the model was not able to detect and misclassified as a false positive. Some reviews which were too short were also misclassified as false positive and in some the model was just not able to get the context of the whole review.

6.2. How to fix this?

One of the ways in which we can improve the performance of the model is by introducing an attention layer before the classification layer which assigns weights to certain rare words which can highly improve the accuracy. Some domain knowledge of the context,tone and intention can be used to inform and guide our models. To keep up with word ambiguities and sarcasm we also need to keep updating our models regularly with new vocabularies and slangs. Training data with curated datasets which are specifically labeled for sarcasm can help the model identify sarcastic patterns. We can also look for linguistic features such as idioms or phrases, hyperbolic language and contradictions to identify sarcasm. We can also use Contextual Embeddings which include models like BERT and RoBERTa which can capture the context of the text in a much better way than the traditional embeddings.Lastly if we want to improve our sentimental analysis we should incorporate contextual sentimental analysis tools which include deep learning models which can better understand the tone and context.

7. Comparison Between biRNN and biLSTM

The first plot shows us that for a given review length how dense it is in the correct predictions. Both the biRNN and biLSTM models gave a similar behaviour that for reviews of length around 50 were the most dense in the correct predictions and too short or too long reviews

were both lesser a part of the correct predictions. Next we analyse the effect of review lengths and rare word occurence on the performance of both the models. We see that as the review length increases the accuracy of both the RNN and LSTM model decreases and if we compare the accuracies of LSTM and RNN the biLSTM model was much better as compared to biRNN. Similar were the observations for the rare word occurences. The biLSTM model performed much better than the biRNN model in terms of accuracy but both show almost similar trend that is with increasing rare words the accuracy of both the biLSTM and biRNN models decrease.

8. Use of AI

The use of AI was restricted to just write certain blocks of code in both the parts. All the concepts and strategies behind it were strictly thought of and implemented by me.