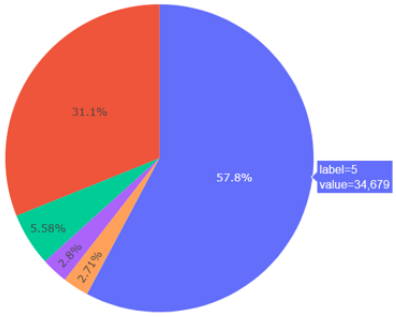
Deep Learning Project Report

Kaggle Competition: Sentiment Analysis of Company Reviews

Submitted By:

Aditya Mishra(M20MA201), Kajal Paradkar(M20MA204), Vandita Agarwal(M20MA208)

DataSet Details: The data set consists of 100,000 reviews of 40 companies, split by Kaggle according to training, public, and private leaderboards, resulting in **training data of 60,000 points** and **test data of 10,000 points**.



Summary of our Project: In this project we have worked on the sentiment analysis of company reviews. The tasks that followed in our project are as follows:

1) Data Preparation: We have done data cleaning, tokenizing the text into words or subwords, and splitting the dataset into training, validation, and test sets.
2) Word embeddings: We have used built word embeddings in our project.
3) Model architecture: We have defined various model architectures which are described below.
4) Training: We have trained our dataset using Adam optimizer and cross entropy loss.
5) Validation: We have used 0.25 percent of data as validation data.
6) Testing: Finally, we predicted ratings on test data.
7) Performance metrics include accuracy, precision, recall, and F1-score.

So, we proceeded with data preprocessing techniques which are, tokenization, encoding, embedding, padding to make it able to pass into a deep learning model. We have used cross-entropy loss and Adam optimizer in our project.

Firstly, we used the CNN model for training company reviews and their ratings.

CNN

The CNN consists of the following things: an embedding layer, a set of convolutional layers with filters of different sizes(3,4,5) to embedding vectors, a max pooling layer, and a fully connected layer.

Vocab_size: 42039, embedding_dim: 500, num_filters: 10, filter_sizes: [3,4,5], num_classes: 5, dropout: 0.2

Then we applied RNN to our dataset.

RNN

I have used single layer and multilayer RNNs, consisting of the following things: an embedding layer, a rnn layer, a fully connected layer.

After that to increase the accuracy, we have introduced LSTM.

LSTM:

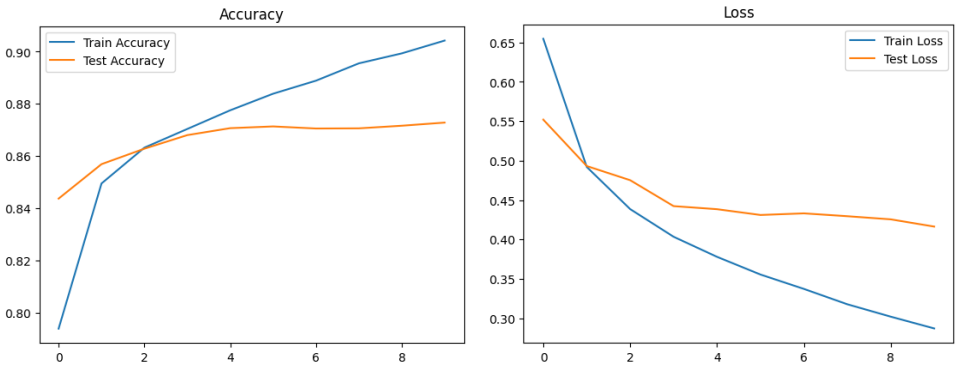
I have used both single layer and multilayer lstms to check the performance of both lstms.
BILSTM:
Then we deployed a BiLSTM model.
BILSTM with Attention:
To check performance of BILSTM with attention, we used this model.
In sequence to it, we finally employed a transformer to our model.

We then compared all these models using various metrics to get best results from these models.

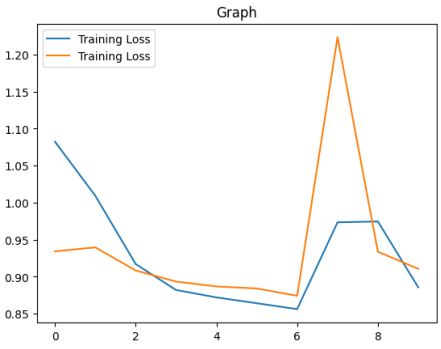
Results

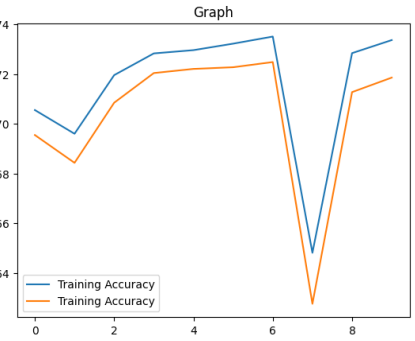
Models	Training Loss	Validation Loss	Training Accuracy	Validation Accuracy	F1 Score	Epochs	Learning Rate
CNN	0.2871	0.416	90.42%	87.28%	0.832	10	0.0001
RNN(single layer)	0.852	0.886	80.02%	72.5%	-	10	0.01
LSTM(single layer)	0.495	0.505	88.06%	85.8%	-	10	0.01
RNN(multilayer)	0.426	0.402	57.6%	57.3%	-	10	0.01
LSTM(multilayer)	0.413	0.47	66.7%	57.2%	-	10	0.01
BiLSTM	0.434	0.469	86.47%	85.82%	0.783	10	0.0001
BiLSTM(attention level)	0.437	0.466	86.48%	86.06%	0.787	10	0.0001
Transformer	0.381	0.479	**91.76%**	**89.13%**	0.86	3	0.00001

CNN

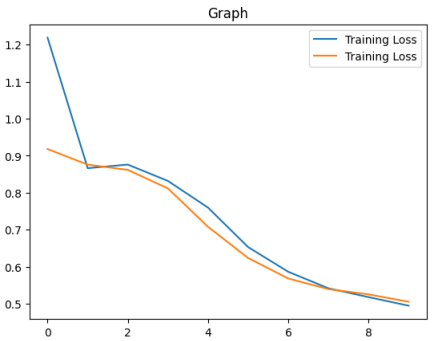


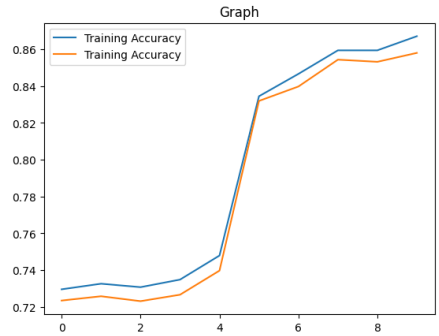
RNN



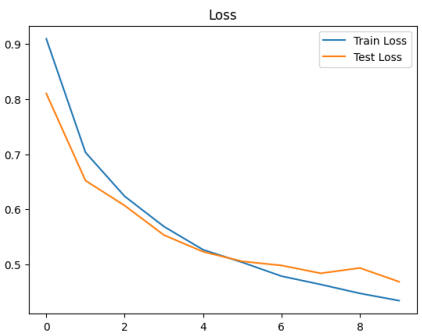


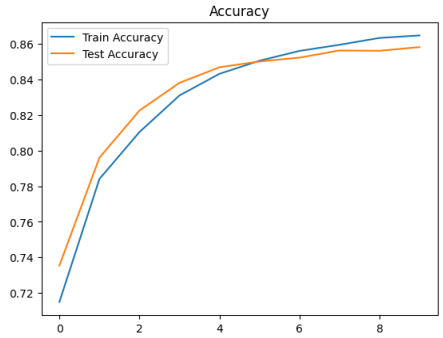
LSTM



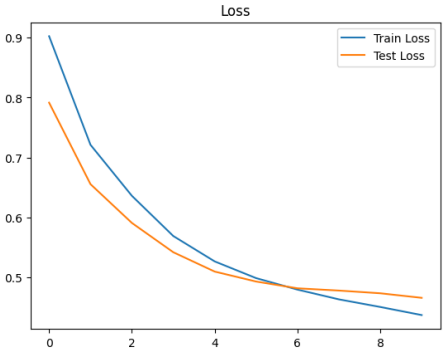


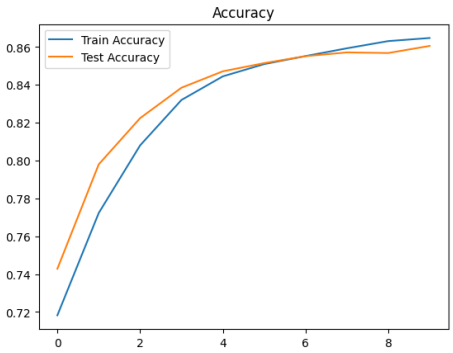
BiLSTM





Bi-LSTM(Attention)





<u>Observations:</u>

1. BiLSTm has given results for only two classes of highest counts. To make this correct, we used attention in BiLSTM that has given results for three classes.
2. Transformer has given the highest accuracy followed by CNN.

References: https://www.kaggle.com/competitions/sentiment-analysis-company-reviews