Hate Intensity Prediction

Team Members: Vinay S, Satya Swaroop G, Aman G  
Team Number: 14

Description:

As part of the project, we have explored various approaches and in this report we walkthrough regarding datasets we used, our approaches, our findings and next steps. We will also recap our committed timelines as reported in project outline.

Approach:

We have identified benchmark datasets for which baselines[1] are available and performed exploratory data anlaysis to identify any patterns and applied tokenization by ensuring we filter noise in the data and finally we experimented with various model architectures by incrementally updating the baseline architecture. The predicted scores from the trained model are evaluated with metrics like Pearson, Cosine Sim, RMSE against baseline results.

## Model Training approaches for hate intensity prediction:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Exp #No | Model architecture | Pearson | Cosine Sim | RMSE |
| 1 | BERT + BiLSTM[1] | 0.787 | 0.975 | 1.3947 |
| 2 | ROBERTA + BiLSTM | 0.818 | 0.98 | 1.1846 |
| 3 | Roberta + RCNN | 0.830 | 0.9799 | 1.1892 |
| 4 | Roberta BiLstm CNN | 0.844 | 0.981 | 1.1517 |

\* RMSE score of baseline is not exactly reproducible as mentioned in the paper and requested authors for the observed variance.

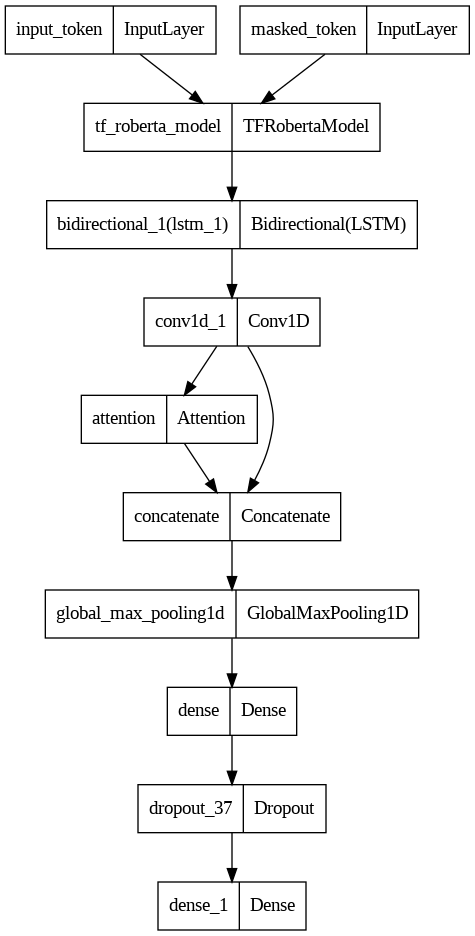
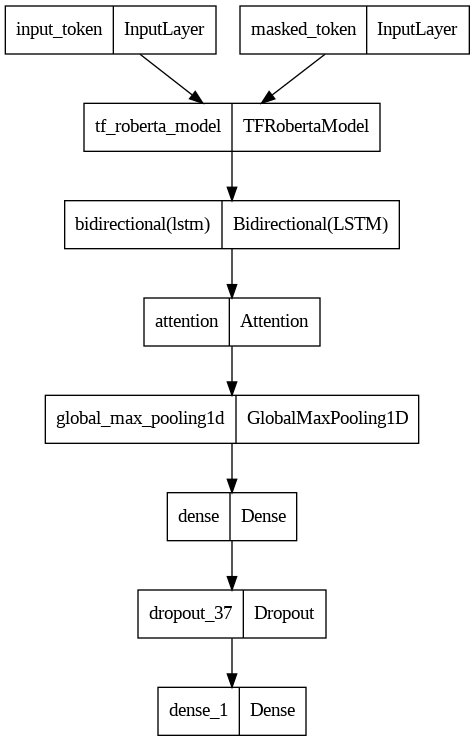
Observations:

1. We are able to significantly improve the baseline performance by changing the embedding from Bert to Roberta representations.
2. We have referred to a similar problem that is “Sarcasm detection”[2] and have implemented similar architecture that is Roberta + RCNN for the current problem statement “Hate Intensity prediction” and results are improving by small fractions.

Hyperparameter tuning trials:

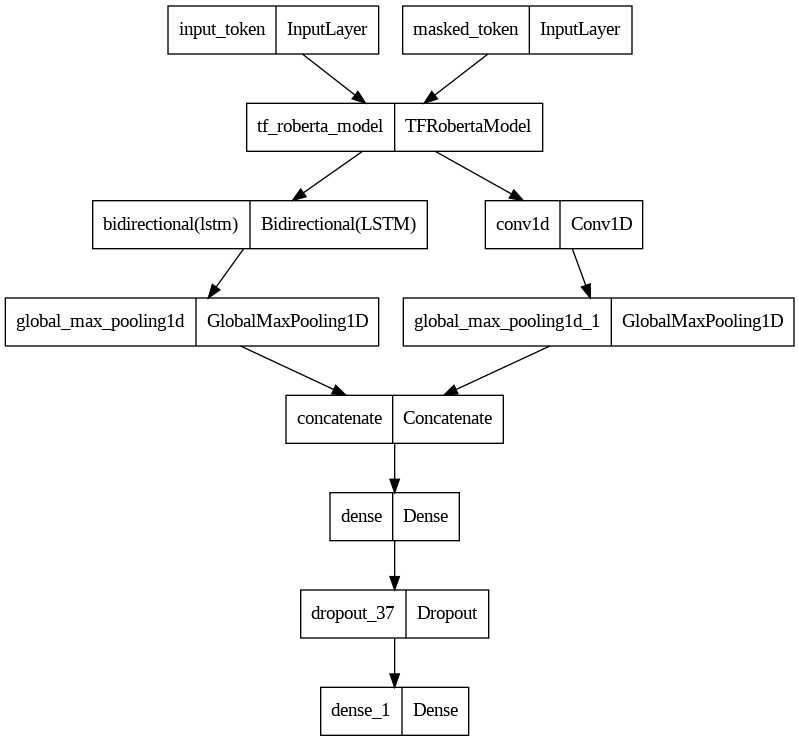
Model Architectures:

Following are the architectures of the experiments:



#2. Roberta + BiLSTM

#3. Roberta + RCNN



#4. Roberta + BiLSTM + CNN

Model description:

**Input Layer**: Takes input\_ids and input\_masks as inputs.

**BERT Embedding Layer**: Transforms the input IDs into embeddings.

**Bi-directional LSTM**: Processes the embeddings and captures sequential information from both directions.

**Attention Layer**: (Optional) Applies self-attention to the LSTM outputs, focusing on different parts of the sequence.

**Global Max Pooling**: Reduces dimensionality by retaining max values from LSTM/Attention outputs.

**Conv1D:** A 1D Convolution layer applies filters on the BERT embeddings, potentially capturing local patterns or n-gram features from the embeddings.

**Concatenation**: The max-pooled outputs of both the LSTM and Conv1D layers are concatenated. This action effectively merges the features learned from both paths.

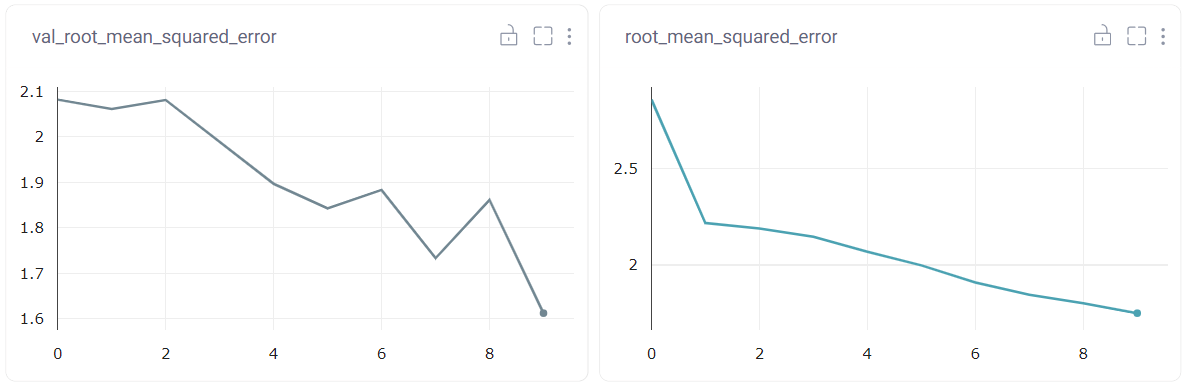
**Dense Layer**: Fully connected layer that can learn representations from the previous layer.

**Dropout**: Reduces overfitting by dropping out nodes during training.

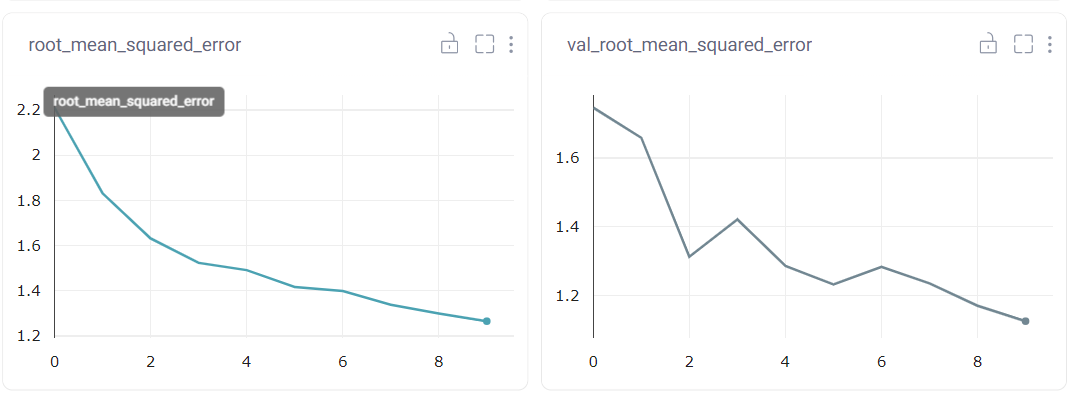
**Output Layer**: Produces the final prediction of hate intensity.

Comet ML experiment tracker Visualizations:

Bert + BiLSTM

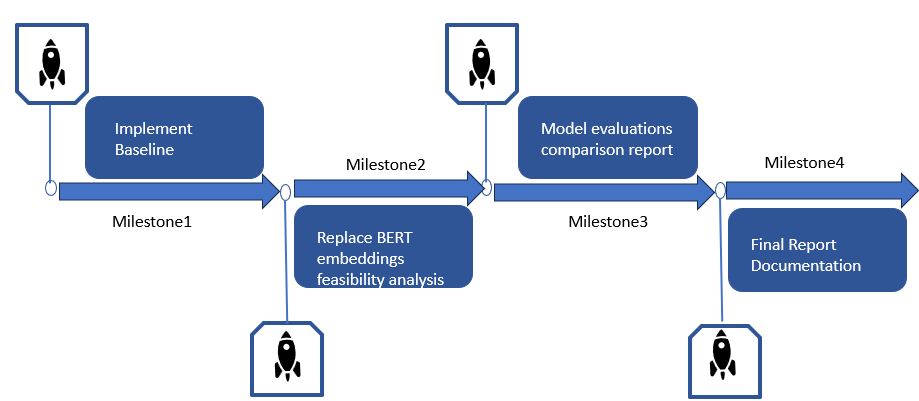


Roberta + BiLSTM + CNN



Revisit Timelines:

We are precisely meeting the timelines as mentioned in the project outline. Where we implemented the baseline and updated the baseline model by modifying the model architecture and able to achieve improved performance than the baseline (Bert + BiLSTM). We have also conducted hyper parameter tuning trials and have explored novel architectures and able to beat the baseline although by few percentages.



Next Steps:

Following are some approaches we will be experimenting with as per feasibility:

* Ensemble Approaches by ensembling the prediction results from various model architectures
* Perform transfer learning by identifying a model that is trained on similar dataset and finetune the last layers specific to the current task.
* Explore deeper representations by replacing Bert with T5 etc based on compute availability.

References:

1. [Hate Intensity Prediction baseline paper](https://browse.arxiv.org/pdf/2206.04007v1.pdf)
2. [A transformer based approach to Irony and Sarcasm detection](https://link.springer.com/article/10.1007/s00521-020-05102-3/tables/3)