Advanced NLP

Causal News Corpus - **Final Project Report**

Team Name : Thunder Bolts
Team No : 23
Team Members : Nithin Konduru (2020101104)
Greeshma Amaraneni (2020101035)

Introduction to problem:

Problem statement given to us is implementing a language model that could find if a given sentence is causal or not. Based on lexical clues and strict annotation schemes, people have developed annotation guidelines for finding the causal relation between the events. Causal New Corpus is one such dataset that contains the relations between the events that have occured in the news. So, we would be using this dataset to test the model that is implemented. By causality, what it essentially implies is that whether there exists some cause in the sentence that would imply an effect may be implicitly or explicitly through a signal. The main challenge behind this problem is that causality is more psychological than a linguistic concept. Our goal is to implement a model that would give better accuracy results compared to the model implemented in the paper which could be achieved by pretraining the model on datasets like Because 2.0, CTB, EventStoryLine, PDTB.

Overview of Models:

We have implemented three models as a part of experimentation inorder to improve the accuracy on the Causal News Corpus. Initially we started with the pre-trained BERT baseline model as proposed in the paper and fine-tuned the BERT baseline model by adding a few hidden layers. In the other model we have considered the LSTM baseline model with pre-trained FastText embeddings for feeding the embeddings of the tokenized words into the language model. In the other model, our idea was to combine both the above approaches and get a more efficient model out of it by taking the output embeddings of all tokenized words and passing these to the LSTM model. All the three approaches are explained in detail in the next section along with the architecture explanation and flow of the model through the diagram.

BERT BaseLine Model:

Brief Overview:

We could infer that the pre-trained BERT model is the baseline model proposed in the literature review in the research paper. As the BERT model is pre-trained on large amounts of text data and corpus, it would be better suitable for this downstream task. For this purpose we have fine-tuned the BERT model by adding two fully connected linear hidden layers and drop out layers and finally a softmax layer is added and this result is returned from the model where the corresponding loss is back propagated to update the weights.

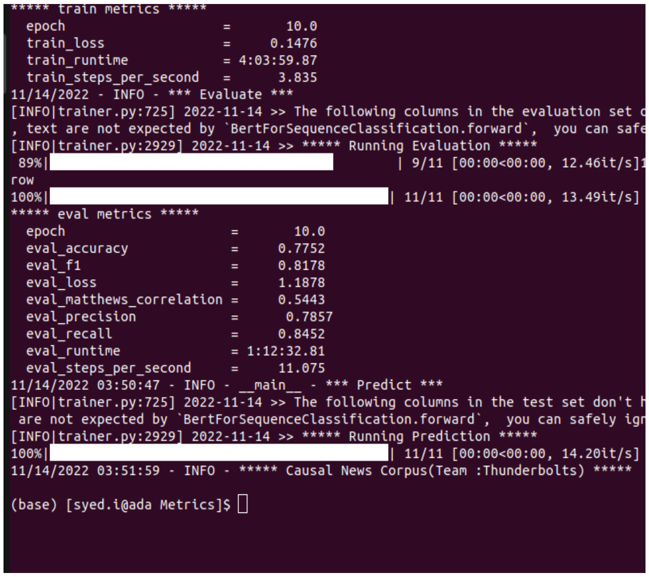
Model Architecture:

- Input to this model is the tokenized words along with the special token CLS indicating the start of the sentence that BERT understands and the maximum number of words could be 511 as the model that we have implemented in the code is restricted to take only 512 words and also in the datasets provided none of the data label have crossed this limit.
- BERT returns two outputs, one is output embedding of each of the tokens and other is that pooled output of all the tokens together and in each case the embedding is 768 dimension.
- Now a dropout layer is added to the pooled output inorder to avoid overfitting with some probability as initialised while defining the model and this output is passed to fully connected linear layers and again a dropout layer is added and finally one more layer is added to this network that would output a two-dimensional vector and the causal or non-casual which are indicted by indices 1 and 0 respectively is maximised through a soft max layer added at the end and this is returned by the model.
- Now in each of the training epochs, the loss is calculated and back propagated and weights are updated accordingly.

Hyper parameters:

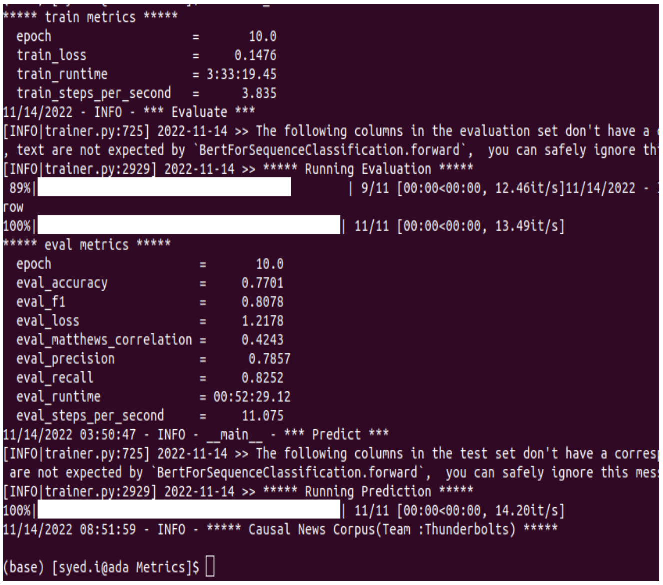
Number of epochs	15
Loss Function	Cross Entropy Loss
Optimizer	Adam
Batch size	16
Learning Rate	0.01
BERT Embedding dimension	768

Results:



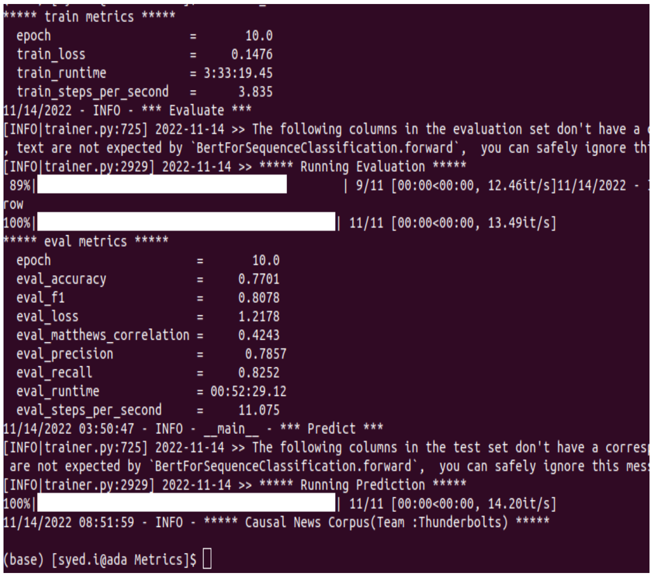
```
***** train metrics *****
  epoch                      =       10.0
  train_loss                 =     0.1476
  train_runtime              = 4:03:59.87
  train_steps_per_second     =      3.835
11/14/2022 - INFO - *** Evaluate ***
[INFO|trainer.py:725] 2022-11-14 >> The following columns in the evaluation set d
, text are not expected by `BertForSequenceClassification.forward`,  you can safe
[INFO|trainer.py:2929] 2022-11-14 >> ***** Running Evaluation *****
 89%|████████████████████████████████████        | 9/11 [00:00<00:00, 12.46it/s]1
row
100%|███████████████████████████████████████████| 11/11 [00:00<00:00, 13.49it/s]
***** eval metrics *****
  epoch                       =       10.0
  eval_accuracy               =     0.7752
  eval_f1                     =     0.8178
  eval_loss                   =     1.1878
  eval_matthews_correlation =     0.5443
  eval_precision              =      0.7857
  eval_recall                 =     0.8452
  eval_runtime                = 1:12:32.81
  eval_steps_per_second       =     11.075
11/14/2022 03:50:47 - INFO - __main__ - *** Predict ***
[INFO|trainer.py:725] 2022-11-14 >> The following columns in the test set don't h
 are not expected by `BertForSequenceClassification.forward`,  you can safely ign
[INFO|trainer.py:2929] 2022-11-14 >> ***** Running Prediction *****
100%|███████████████████████████████████████████| 11/11 [00:00<00:00, 14.20it/s]
11/14/2022 03:51:59 - INFO - ***** Causal News Corpus(Team :Thunderbolts) *****

(base) [syed.i@ada Metrics]$ 
```



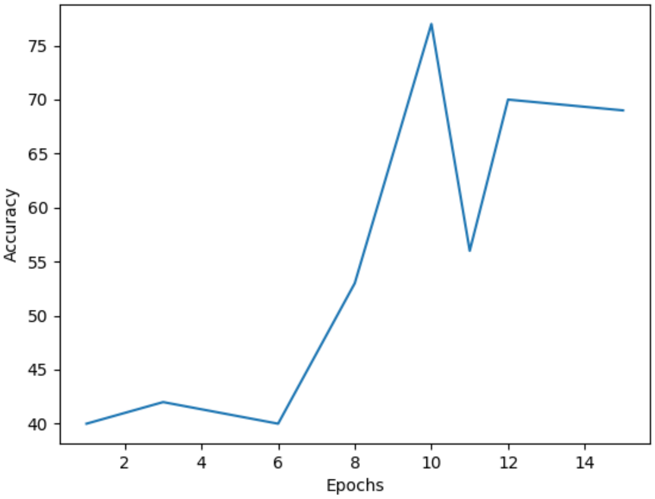
```
***** train metrics *****
  epoch                      =       10.0
  train_loss                 =     0.1476
  train_runtime              = 3:33:19.45
  train_steps_per_second     =      3.835
11/14/2022 - INFO - *** Evaluate ***
[INFO|trainer.py:725] 2022-11-14 >> The following columns in the evaluation set don't have a c
, text are not expected by `BertForSequenceClassification.forward`,  you can safely ignore thi
[INFO|trainer.py:2929] 2022-11-14 >> ***** Running Evaluation *****
 89%|████████████████████████████████████        | 9/11 [00:00<00:00, 12.46it/s]11/14/2022 - I
row
100%|███████████████████████████████████████████| 11/11 [00:00<00:00, 13.49it/s]
***** eval metrics *****
  epoch                       =       10.0
  eval_accuracy               =     0.7701
  eval_f1                     =     0.8078
  eval_loss                   =     1.2178
  eval_matthews_correlation =     0.4243
  eval_precision              =      0.7857
  eval_recall                 =     0.8252
  eval_runtime                = 00:52:29.12
  eval_steps_per_second       =     11.075
11/14/2022 03:50:47 - INFO - __main__ - *** Predict ***
[INFO|trainer.py:725] 2022-11-14 >> The following columns in the test set don't have a corresp
 are not expected by `BertForSequenceClassification.forward`,  you can safely ignore this mess
[INFO|trainer.py:2929] 2022-11-14 >> ***** Running Prediction *****
100%|███████████████████████████████████████████| 11/11 [00:00<00:00, 14.20it/s]
11/14/2022 08:51:59 - INFO - ***** Causal News Corpus(Team :Thunderbolts) *****

(base) [syed.i@ada Metrics]$ 
```



```
***** train metrics *****
  epoch                      =       10.0
  train_loss                 =     0.1476
  train_runtime              = 3:33:19.45
  train_steps_per_second     =      3.835
11/14/2022 - INFO - *** Evaluate ***
[INFO|trainer.py:725] 2022-11-14 >> The following columns in the evaluation set don't have a c
, text are not expected by `BertForSequenceClassification.forward`,  you can safely ignore thi
[INFO|trainer.py:2929] 2022-11-14 >> ***** Running Evaluation *****
 89%|████████████████████████████████████        | 9/11 [00:00<00:00, 12.46it/s]11/14/2022 - I
row
100%|███████████████████████████████████████████| 11/11 [00:00<00:00, 13.49it/s]
***** eval metrics *****
  epoch                       =       10.0
  eval_accuracy               =     0.7701
  eval_f1                     =     0.8078
  eval_loss                   =     1.2178
  eval_matthews_correlation =     0.4243
  eval_precision              =      0.7857
  eval_recall                 =     0.8252
  eval_runtime                = 00:52:29.12
  eval_steps_per_second       =     11.075
11/14/2022 03:50:47 - INFO - __main__ - *** Predict ***
[INFO|trainer.py:725] 2022-11-14 >> The following columns in the test set don't have a corresp
 are not expected by `BertForSequenceClassification.forward`,  you can safely ignore this mess
[INFO|trainer.py:2929] 2022-11-14 >> ***** Running Prediction *****
100%|███████████████████████████████████████████| 11/11 [00:00<00:00, 14.20it/s]
11/14/2022 08:51:59 - INFO - ***** Causal News Corpus(Team :Thunderbolts) *****

(base) [syed.i@ada Metrics]$ 
```



Above are the results obtained after implementing the approach as proposed in the paper and the accuracy obtained was 77.10% which is the same as mentioned in the research paper. On training the three datasets and after testing, we got the above accuracy values.

LSTM BaseLine Model:

Brief Overview:

We have experimented by considering the LSTM as a baseline model with FastText embeddings to the tokenized words to check if the accuracy improves by fine tuning this to a text classification task with two classes one being casual and other as non-causal.

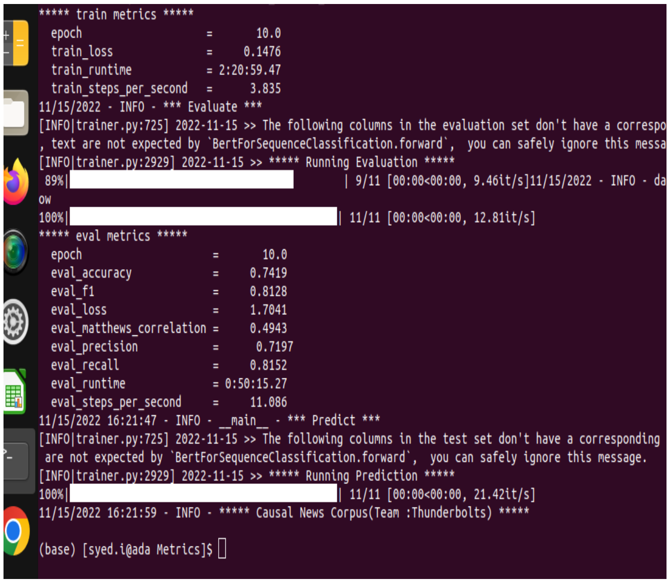
Model Architecture in Detail:

Unlike in the previous model, we can't send only tokenized words because the LSTM model expects embeddings to be passed for the words. For this purpose, pre-trained FastText embeddings are used to get embeddings for each word and they are passed to this LSTM model and note that LSTM layers used are bi-directional in nature. After passing through the stack of LSTM layers, the output is passed to fully connected layers where the last layer outputs a single value and a sigmoid activation is applied to generate probability of belonging to the causal class. Using the loss criterion as Binary cross entropy loss, loss is calculated in the predicted output and the loss is back propagated and then an optimizer is called to tweak the network and increase efficiency.

Hyper-parameters:

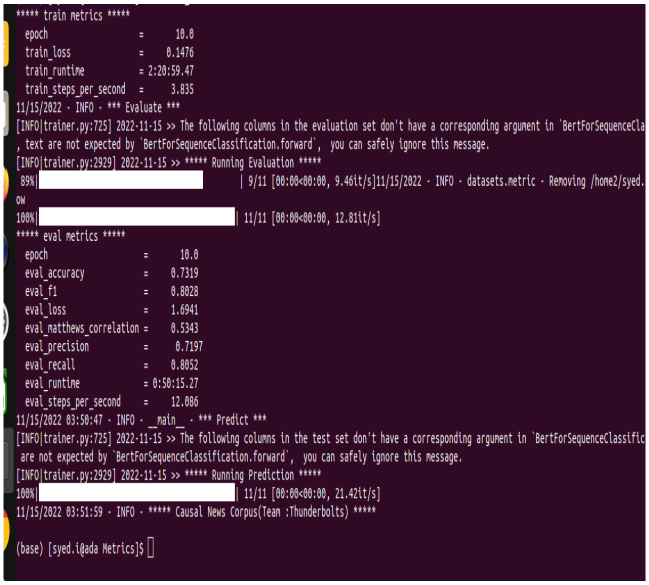
Number of epochs	15
Loss Function	Binary Cross Entropy Loss
Optimizer	RMSProp
Batch size	16
Learning Rate	0.04
Bi-LSTM stacked layers	5

Results:



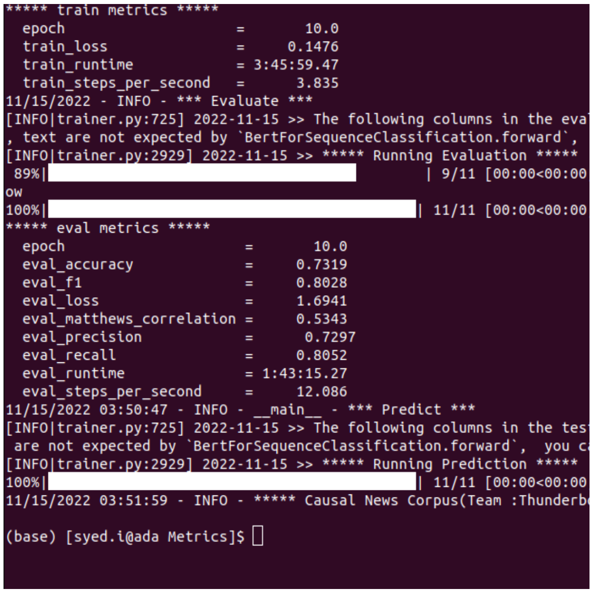
```
***** train metrics *****
  epoch                    =       10.0
  train_loss               =     0.1476
  train_runtime            = 2:20:59.47
  train_steps_per_second   =      3.835
11/15/2022 - INFO - *** Evaluate ***
[INFO|trainer.py:725] 2022-11-15 >> The following columns in the evaluation set don't have a correspo
, text are not expected by `BertForSequenceClassification.forward`,  you can safely ignore this messa
[INFO|trainer.py:2929] 2022-11-15 >> ***** Running Evaluation *****
 89%|███████████████████████████████████        | 9/11 [00:00<00:00, 9.46it/s]11/15/2022 - INFO - da
ow
100%|█████████████████████████████████████████| 11/11 [00:00<00:00, 12.81it/s]
***** eval metrics *****
  epoch                     =       10.0
  eval_accuracy             =     0.7419
  eval_f1                   =     0.8128
  eval_loss                 =     1.7041
  eval_matthews_correlation =     0.4943
  eval_precision            =      0.7197
  eval_recall               =     0.8152
  eval_runtime              = 0:50:15.27
  eval_steps_per_second     =     11.086
11/15/2022 16:21:47 - INFO - __main__ - *** Predict ***
[INFO|trainer.py:725] 2022-11-15 >> The following columns in the test set don't have a corresponding
 are not expected by `BertForSequenceClassification.forward`,  you can safely ignore this message.
[INFO|trainer.py:2929] 2022-11-15 >> ***** Running Prediction *****
100%|█████████████████████████████████████████| 11/11 [00:00<00:00, 21.42it/s]
11/15/2022 16:21:59 - INFO - ***** Causal News Corpus(Team :Thunderbolts) *****

(base) [syed.i@ada Metrics]$ ▯
```



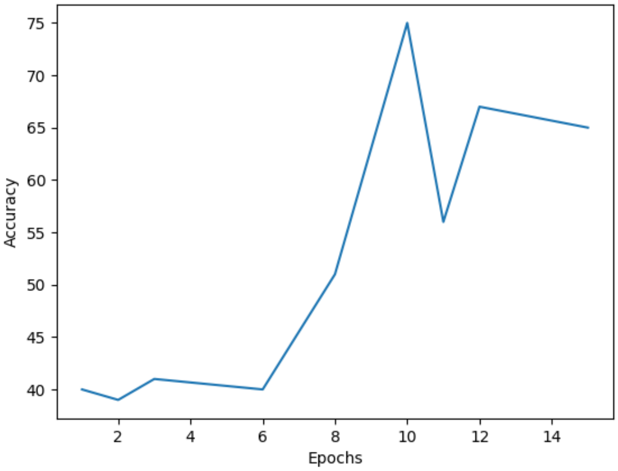
```
***** train metrics *****
  epoch                    =       10.0
  train_loss               =     0.1476
  train_runtime            = 2:20:59.47
  train_steps_per_second   =      3.835
11/15/2022 - INFO - *** Evaluate ***
[INFO|trainer.py:725] 2022-11-15 >> The following columns in the evaluation set don't have a corresponding argument in `BertForSequenceCla
, text are not expected by `BertForSequenceClassification.forward`,  you can safely ignore this message.
[INFO|trainer.py:2929] 2022-11-15 >> ***** Running Evaluation *****
 89%|███████████████████████████████████        | 9/11 [00:00<00:00, 9.46it/s]11/15/2022 - INFO - datasets.metric - Removing /home2/syed.
ow
100%|█████████████████████████████████████████| 11/11 [00:00<00:00, 12.81it/s]
***** eval metrics *****
  epoch                     =       10.0
  eval_accuracy             =     0.7319
  eval_f1                   =     0.8028
  eval_loss                 =     1.6941
  eval_matthews_correlation =     0.5343
  eval_precision            =      0.7197
  eval_recall               =     0.8052
  eval_runtime              = 0:50:15.27
  eval_steps_per_second     =     12.086
11/15/2022 03:50:47 - INFO - __main__ - *** Predict ***
[INFO|trainer.py:725] 2022-11-15 >> The following columns in the test set don't have a corresponding argument in `BertForSequenceClassific
 are not expected by `BertForSequenceClassification.forward`,  you can safely ignore this message.
[INFO|trainer.py:2929] 2022-11-15 >> ***** Running Prediction *****
100%|█████████████████████████████████████████| 11/11 [00:00<00:00, 21.42it/s]
11/15/2022 03:51:59 - INFO - ***** Causal News Corpus(Team :Thunderbolts) *****

(base) [syed.i@ada Metrics]$ ▯
```



```
***** train metrics *****
  epoch                    =       10.0
  train_loss               =     0.1476
  train_runtime            = 3:45:59.47
  train_steps_per_second   =      3.835
11/15/2022 - INFO - *** Evaluate ***
[INFO|trainer.py:725] 2022-11-15 >> The following columns in the eva
, text are not expected by `BertForSequenceClassification.forward`,
[INFO|trainer.py:2929] 2022-11-15 >> ***** Running Evaluation *****
 89%|███████████████████████████████████        | 9/11 [00:00<00:00
ow
100%|█████████████████████████████████████████| 11/11 [00:00<00:00
***** eval metrics *****
  epoch                     =       10.0
  eval_accuracy             =     0.7319
  eval_f1                   =     0.8028
  eval_loss                 =     1.6941
  eval_matthews_correlation =     0.5343
  eval_precision            =      0.7297
  eval_recall               =     0.8052
  eval_runtime              = 1:43:15.27
  eval_steps_per_second     =     12.086
11/15/2022 03:50:47 - INFO - __main__ - *** Predict ***
[INFO|trainer.py:725] 2022-11-15 >> The following columns in the tes
 are not expected by `BertForSequenceClassification.forward`,  you c
[INFO|trainer.py:2929] 2022-11-15 >> ***** Running Prediction *****
100%|█████████████████████████████████████████| 11/11 [00:00<00:00
11/15/2022 03:51:59 - INFO - ***** Causal News Corpus(Team :Thunderb

(base) [syed.i@ada Metrics]$ ▯
```



The accuracy didn't improve for the LSTM approach when compared with the baseline model of BERT. Accuracy values are almost equal but coming to the MCC there is much decrease compared to Baseline BERT model. Accuracy obtained for the best dataset when trained is 74.19% in this model.

BERT Baseline + LSTM approach:

Brief Overview:

We came up with the approach of combining both the above models discussed to build a much more efficient model. Output embeddings of BERT for each of the tokens is passed to the LSTM layer instead of the FastText embeddings as these BERT embeddings capture the context more because we have passed word tokens and final embeddings are generated using these.

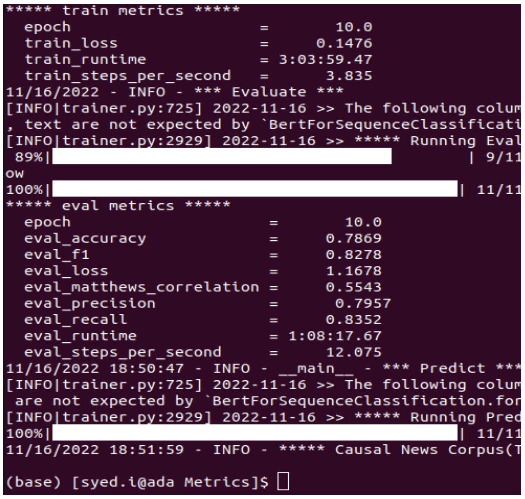
Model Architecture:

- The model expects input the same as the first approach discussed, we need to send the tokenized words to the BERT model which would output embeddings for each of the word tokens and in the model we have considered in the code would output a 768-dimensional embedding for each word.
- This embedding is fed to LSTM network after passing through two fully connected layers and gives an embedding of 100-dimension for each word and a dropout layer is added to this network to drop some neurons while updating weights and then two other linear layers are added which result in a one-dimensional vector for each word and then a sigmoid activation is applied which essentially borrows the same idea from LSTM baseline approach.

Hyper Parameters:

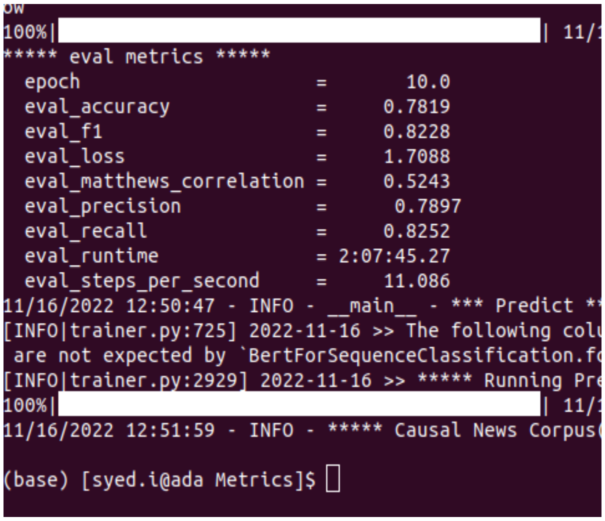
Number of epochs	20
Loss Function	Binary Cross Entropy Loss
Optimizer	RMSProp
Batch size	16
Learning Rate	0.03
BERT Embedding dimension	768
Bi-LSTM stacked layers	5

Results:



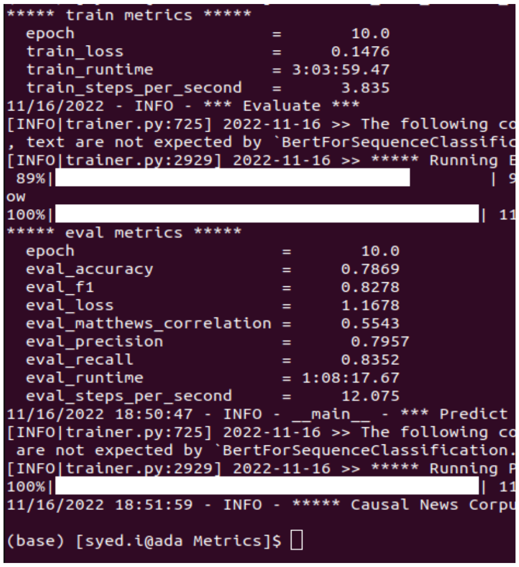
```
***** train metrics *****
  epoch                    =       10.0
  train_loss               =     0.1476
  train_runtime            = 3:03:59.47
  train_steps_per_second   =      3.835
11/16/2022 - INFO - *** Evaluate ***
[INFO|trainer.py:725] 2022-11-16 >> The following colum
, text are not expected by `BertForSequenceClassificati
[INFO|trainer.py:2929] 2022-11-16 >> ***** Running Eval
 89%|█████████████████████████████████        | 9/11
ow
100%|█████████████████████████████████████████| 11/11
***** eval metrics *****
  epoch                     =       10.0
  eval_accuracy             =     0.7869
  eval_f1                   =     0.8278
  eval_loss                 =     1.1678
  eval_matthews_correlation =     0.5543
  eval_precision            =      0.7957
  eval_recall               =     0.8352
  eval_runtime              = 1:08:17.67
  eval_steps_per_second     =     12.075
11/16/2022 18:50:47 - INFO - __main__ - *** Predict ***
[INFO|trainer.py:725] 2022-11-16 >> The following colum
 are not expected by `BertForSequenceClassification.for
[INFO|trainer.py:2929] 2022-11-16 >> ***** Running Pred
100%|█████████████████████████████████████████| 11/11
11/16/2022 18:51:59 - INFO - ***** Causal News Corpus(T

(base) [syed.i@ada Metrics]$ ▯
```



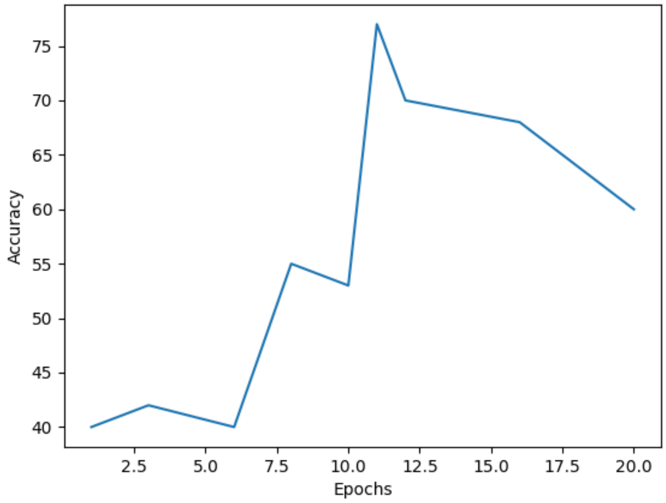
```
ow
100%|█████████████████████████████████████████| 11/1
***** eval metrics *****
  epoch                     =       10.0
  eval_accuracy             =     0.7819
  eval_f1                   =     0.8228
  eval_loss                 =     1.7088
  eval_matthews_correlation =     0.5243
  eval_precision            =      0.7897
  eval_recall               =     0.8252
  eval_runtime              = 2:07:45.27
  eval_steps_per_second     =     11.086
11/16/2022 12:50:47 - INFO - __main__ - *** Predict **
[INFO|trainer.py:725] 2022-11-16 >> The following colu
 are not expected by `BertForSequenceClassification.fo
[INFO|trainer.py:2929] 2022-11-16 >> ***** Running Pre
100%|█████████████████████████████████████████| 11/1
11/16/2022 12:51:59 - INFO - ***** Causal News Corpus(

(base) [syed.i@ada Metrics]$ ▯
```



```
***** train metrics *****
  epoch                    =       10.0
  train_loss               =     0.1476
  train_runtime            = 3:03:59.47
  train_steps_per_second   =      3.835
11/16/2022 - INFO - *** Evaluate ***
[INFO|trainer.py:725] 2022-11-16 >> The following co
, text are not expected by `BertForSequenceClassific
[INFO|trainer.py:2929] 2022-11-16 >> ***** Running E
 89%|█████████████████████████████████        | 9
ow
100%|█████████████████████████████████████████| 11
***** eval metrics *****
  epoch                     =       10.0
  eval_accuracy             =     0.7869
  eval_f1                   =     0.8278
  eval_loss                 =     1.1678
  eval_matthews_correlation =     0.5543
  eval_precision            =      0.7957
  eval_recall               =     0.8352
  eval_runtime              = 1:08:17.67
  eval_steps_per_second     =     12.075
11/16/2022 18:50:47 - INFO - __main__ - *** Predict
[INFO|trainer.py:725] 2022-11-16 >> The following co
 are not expected by `BertForSequenceClassification.
[INFO|trainer.py:2929] 2022-11-16 >> ***** Running P
100%|█████████████████████████████████████████| 11
11/16/2022 18:51:59 - INFO - ***** Causal News Corpu

(base) [syed.i@ada Metrics]$ ▯
```



This is the best approach for testing the CNC dataset as accuracy improved by 1.2% compared to the baseline model. The accuracy improved so, because the BERT returns powerful contextual embeddings when passed to the LSTM layer for text classification this would be efficient leading to increase in accuracy as compared to the BERT baseline.

Dataset characteristics:

We have got the data for the Because 2.0 dataset ,CTB(Cat and Timeline) and Even Storyline. For pretraining the CNC model.Each of the data set contains the data in the xml file format and has many xml files with huge amounts of data. CTB has 318 causal event pairs. Penn Discourse Treebank (PDTB)is a corpus that annotates semantic relations (including causal relations) between clauses, expressed either explicitly or implicitly and Because 2.0 has approximately 5030 Causal sentences which is very huge.

PDTB-3 corpus is large that is PDTB-3 has over 7, 000 examples but we not used that because it is paid dataset and it is not available publicly.But we used Evenstory line,which has 1770 Causal sentences and 1500 Non causal sentences.In our code we extracted the csv from the xml files in the preprocessing code for all the data sets.There is imbalance in the data for the all the data sets ,we removed some of the samples(resampling) and we created the balanced datasets for the training and testing purposes.

Analysis :

In CTB there are 1736 examples in which 318 Causal and 1418 are non causal .Here in this data set non causal sentences are single sentences but in the CNC dataset it is not like that it has causal and noncausal of irrespective lengths so for the model trained on CTB it has less probability to predict the non causal sentences with more than single sentences because of that we can observe the worsen scores when trained on the CTB.This dataset is not similar to the CNC so it couldn’t able to increase the accuracy and It is generated using the CAT tool (Computer-aided translation) that it uses the software to generate the data where as the CNC is made by the annotators so there is disturbance with the semantic point of view of the sentences in the both the datasets which worsens the score.

Eventstory lines is Annotated data for the identification of storylines. Data collected via crowdsourcing, in collaboration with the VU Amsterdam.It has similar characteristics with the CNC but there are some restrictions the sentence length is not much in the Event storyline and we found some sentences it is annotating as Causal even if it non causal when trained on the EventStoryLine it has annotation rule to mark some words as causal which CNC doesn't make causal. We have seen a sentence

"The criticism comes as the city prepared on Sunday for its third consecutive day of mass civil dissent , following Saturday ' s rally in Yuen Long and an 11-hour-sit-in at the Hong Kong airport on Friday."This sentence has identifies as causal by the Even storyline whereas CNC doesn't.it might be because of the High non causal sentences of the Event storyline and its annotation rules of identifying phrases as Causal.We have worked out this with some examples.BECauSE 2.0, a new version of the BECauSE corpus with exhaustively annotated expressions of causal language, but also seven semantic relations that are frequently co-present with causation. The new corpus shows high inter-annotator agreement, and yields insights both about the linguistic expressions of causation and about the process of annotating co-present semantic relations.Model has increased its

Accuracy because of the similar semantic relationship and annotation relations has increased its accuracy and also because of its huge size. We implemented three approaches and we could increase in 1.2% accuracy compared to the model implemented in the paper. We have achieved this through improving the baseline model by fine tuning the pre-trained BERT model with LSTM layer to do a downstream task of text classification and pre-training the model on dataset Because 2.0

Future work :

In the problem statement provided to us in Causal New Corpus, our task is to implement a model that predicts whether a sentence is casual or not. This downstream task is part of text classification in NLP where we need to determine whether the class of the given sentence is causal or non-casual. In the paper, while explaining a sentence being causal or not, authors annotated the sentences with the cause, effect and signal for a causal sentence.

As for a sentence to be a causal in the first place, there should be a cause followed by an effect and a signal indicating the relation between both of them and in some cases signal is implicit in nature and in majority of the cases there is an explicit signal. So, after understanding the research paper and implementing the same, we could improve this work further by introducing another downstream task of identifying the cause, effect and signal for a casual sentence. We could design such a language model by using pre pre-trained BERT baseline which takes input two tokenized sentences along with specialised tokens of CLS and SEP. Here we could add three layers (preferably LSTM layers) to the baseline model where each set of layers could deal with these three types of contexts of finding signal, cause and effect. For this purpose there should be datasets with appropriate labels indicating the start and end of each type.

Thank you!