**Technical** **Report**

**Model Name: Bert-Base cased model**

**Bert model performance without tuning techniques**

The accuracy is around 88 % and eval loss, Precision, recall,f1\_score as mentioned below.

* Optimizer used = Adam
* Epochs = 2 [GPU support]
* Arguments used = Default Arguments from Pre-trained NER model
* Working Performance for test\_data =

{'eval\_loss': 0.15920350193563435,

'precision': 0.8308423913043478,

'recall': 0.7837612148988231,

'f1\_score': 0.8066153692415309}

**Reason for less performance**

* Default Arguments
* Less number of epochs for training model
* Due to a smaller number of propagations vanishing gradient Problem takes place

**Bert model Performance with tuning techniques**

'eval\_loss': 0.10674857721788893,

'f1\_score': 0.8585737947440074,

'precision': 0.8754974189344961,

'recall': 0.8422920403479334

**Reason for good performance**

* The accuracy is around 94 and eval loss, Precision, recall,f1\_score as mentioned
* Optimizers used = Adam
* Epochs= 3
* Arguments used = selected best Arguments which makes model to perform better
* Use of selecting best Arguments

1. We can select proper learning rate suitable for the model( based on data we have )
2. Selecting Batch size for both training part and test part (decides how many samples should pass for once epoch
3. We can select the length of the sentence for our data which makes model to train on those sentence\_length
4. We can decide the word should be in lower case or upper case
5. Selecting the sentence length and word representation makes model to learn from the data
6. Selecting best optimizer also makes the model to perform well (how best optimizer helps us)
7. If we use adam optimizer we can change the learning rate for each epoch

**Model Name = distilbert-base-uncased**

The accuracy is around 82 % and eval loss, Precision, recall,f1\_score as mentioned below.

* Optimizer used = Adam
* Epochs = 2
* Arguments used = Default Arguments distilbert model
* Working Performance for test\_data =

**Reason for less performance**

* Default Arguments
* Less number of epochs for training model
* Due to a smaller number of propagations vanishing gradient Problem takes place and High epochs takes exploding gradient problem.

'eval\_loss': 0.4154712584822213,

'f1\_score': 0.29140490811980896,

'precision': 0.3696132266921332,

'recall': 0.2405134578785878

* Distil Bert giving less performance not only because of default arguments but also due to model architecture
* Distil BERT is a small, fast, cheap and light . It has 40% less parameters than Bert
* 6-layer, 768-hidden, 12-heads, 65M parameters The DistilBERT model distilled from the BERT model *bert-base-cased* checkpoint, with an additional question answering layer.
* But for Bert model we are having 12-layer, 768-hidden, 12-heads, 109M parameters.

**Bidirectional LSTM**

* Simple LSTM RNN model was not giving better performance
* But Bidirectional lstm was able to perform better than LSTM RNN
* Working well with training data accuracy is almost around 96
* Number of propagations used = 3
* Loss is also less

**Reasons for Overfitting Problem**

* But there is an overfitting problem
* Model is working well with training data and not giving good results with test data
* Main reason for over fitting problem is not following attention mechanism which follows seq to seq .
* But when using Bert and other models which follows encoder and decoder layers with huge parameters. The model can easily understand the importance of the token based of their probability value