2-

comparison of both the models(Indic-BERT and Indic-NER)

Macro-F1 score on the 20% test set -

Macro-F1 Score(Indic-BERT) = 0.698273640814223
Macro-F1 Score(Indic-NER) = 0.7997017892644135

As we can see from the above values of Macro-F1 Scores that Indic-NER performed better than Indic-BERT on the task of NER.Since, both the models have been trained on the same dataset keeping the hyper parameters same, the difference arises due to the fact that BERT (Bidirectional Encoder Representations from Transformers) relies on bidirectional context representations, but the token embeddings and context window will not capture the specific linguistic nuances of Indic languages as effectively as Indic-NER as Indic-NER incorporates specific positional encodings or modifications to better represent the linguistic structure of Indic languages.

validation macro-F1 scores -

eval_overall_f1(Indic-NER) = 0.7761
eval_overall_f1(Indic-BERT) = 0.6717

4-

Precision, Recall, and Macro-F1-score.

Indic-NER

```
Precision of  B-PER  -  0.8181818181818182
Recall of  B-PER  -  0.8181818181818182
F-Score of  B-PER  -  0.8181818181818182
Precision of  I-PER  -  0.6666666666666666
Recall of  I-PER  -  0.6666666666666666
F-Score of  I-PER  -  0.6666666666666666
Precision of  B-LOC  -  0.6
Recall of  B-LOC  -  0.6
F-Score of  B-LOC  -  0.6
Precision of  I-LOC  -  0.0
Recall of  I-LOC  -  0.0
F-Score of  I-LOC  -  0
Precision of  B-ORG  -  0.36363636363636365
Recall of  B-ORG  -  0.36363636363636365
F-Score of  B-ORG  -  0.36363636363636365
Precision of  I-ORG  -  0.6666666666666666
Recall of  I-ORG  -  0.6666666666666666
F-Score of  I-ORG  -  0.6666666666666666
Precision of  B-MISC  -  0.0
Recall of  B-MISC  -  0.0
F-Score of  B-MISC  -  0
Precision of  I-MISC  -  0.0
Recall of  I-MISC  -  0.0
F-Score of  I-MISC  -  0
Precision of  Other  -  0.9607476635514018
Recall of  Other  -  0.9607476635514018
F-Score of  Other  -  0.9607476635514018
Macro_F1_Score -  0.45287768652254634
```

Indic-BERT

```
Precision of  B-PER  -  0.272727
Recall of  B-PER  -  0.272727272727272
F-Score of  B-PER  -  0.2727272727272727
Precision of  I-PER  -  0.16666666666666666
Recall of  I-PER  -  0.16666666666666666
F-Score of  I-PER  -  0.16666666666666666
Precision of  B-LOC  -  0.0
Recall of  B-LOC  -  0.0
F-Score of  B-LOC  -  0
Precision of  I-LOC  -  0.0
Recall of  I-LOC  -  0.0
F-Score of  I-LOC  -  0
Precision of  B-ORG  -  0.0
Recall of  B-ORG  -  0.0
F-Score of  B-ORG  -  0
Precision of  I-ORG  -  0.0
Recall of  I-ORG  -  0.0
F-Score of  I-ORG  -  0
Precision of  B-MISC  -  0.0
Recall of  B-MISC  -  0.0
F-Score of  B-MISC  -  0
Precision of  I-MISC  -  0.0
Recall of  I-MISC  -  0.0
F-Score of  I-MISC  -  0
Precision of  Other  -  0.9906542056074766
Recall of  Other  -  0.9906542056074766
F-Score of  Other  -  0.9906542056074766
Macro_F1_Score -  0.15889423833349067
```

Notebook editor cells

ChatGPT

```
Precision of  B-PER  -  0.9090909090909091
Recall of  B-PER  -  0.9090909090909091
F-Score of  B-PER  -  0.9090909090909091
Precision of  I-PER  -  0.8333333333333334
Recall of  I-PER  -  0.8333333333333334
F-Score of  I-PER  -  0.8333333333333334
Precision of  B-LOC  -  1.0
Recall of  B-LOC  -  1.0
F-Score of  B-LOC  -  1.0
Precision of  I-LOC  -  1.0
Recall of  I-LOC  -  1.0
F-Score of  I-LOC  -  1.0
Precision of  B-ORG  -  0.09090909090909091
Recall of  B-ORG  -  0.09090909090909091
F-Score of  B-ORG  -  0.09090909090909091
Precision of  I-ORG  -  0.0
Recall of  I-ORG  -  0.0
F-Score of  I-ORG  -  0
Precision of  B-MISC  -  0.0
Recall of  B-MISC  -  0.0
F-Score of  B-MISC  -  0
Precision of  I-MISC  -  0.0
Recall of  I-MISC  -  0.0
F-Score of  I-MISC  -  0
Precision of  Other  -  0.994392523364486
Recall of  Other  -  0.994392523364486
F-Score of  Other  -  0.994392523364486
Macro_F1_Score -  0.5364139840775355
```

Above given are the Precision,Recall and F1-Scores of each class with respect to the model.Indic-NER as well as Indic-BERT don’t have MISC class and therefore all the MISC values are 0.Some of the other tags also have 0 values as the 25 sentences do not contains any instance of such tag.(For example - I-LOC)

5-

Values of hyper-parameters chosen to be tuned are Train-Batch Size,Eval-Batch Size and Learning Rate.

Significance of the Hyper Parameters -

Train-Batch Size - Smaller batch sizes can introduce more noise during training, as the model's parameters are updated more frequently. On the other hand, larger batch sizes might lead to a smoother convergence but may have a regularising effect, potentially affecting the model's ability to generalise to unseen data.Smaller batch sizes may lead to faster convergence, but they may also result in more oscillations in the loss function.

Eval-Batch Size - The evaluation batch size can influence the speed and efficiency of the model evaluation process.Larger batch sizes may speed up evaluation but could require more memory. Smaller batch sizes may be computationally lighter but might take longer to process all evaluation samples.The choice of Eval-Batch Size affects the assessment of the model's generalisation performance on new, unseen data.

Learning Rate - The learning rate determines how quickly or slowly a model converges during training. A too-high learning rate might cause the model to overshoot the optimal parameters, leading to divergence or oscillation, while a too-low learning rate may result in slow convergence.The learning rate can impact the generalisation ability of the model. If the learning rate is too high, the model may memorise the training data and a too-low learning rate might result in underfitting.

Optimal values-

Optimal results obtained for following arguments: (for both the models)

- per_device_train_batch_size - 8
- Per_device_eval_batch_size - 8
- Learning_rate - 5e-5

Output:

Indic-BERT -

(i)

```
# args=TrainingArguments(output_dir='output_dir',max_steps=5)
args=TrainingArguments(
    output_dir='output_dir',
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3)
```

```
***** eval metrics *****
  epoch                   =        3.0
  eval_LOC_f1             =     0.7294
  eval_LOC_number         =      10213
  eval_LOC_precision      =     0.7255
  eval_LOC_recall         =     0.7334
  eval_ORG_f1             =      0.563
  eval_ORG_number         =       9786
  eval_ORG_precision      =     0.5725
  eval_ORG_recall         =     0.5539
  eval_PER_f1             =     0.7151
  eval_PER_number         =      10568
  eval_PER_precision      =     0.7233
  eval_PER_recall         =     0.7071
  eval_loss               =     0.2583
  eval_overall_accuracy   =     0.9215
  eval_overall_f1         =     0.6717
  eval_overall_precision  =     0.6767
  eval_overall_recall     =     0.6668
  eval_runtime            = 0:04:13.47
  eval_samples_per_second =     53.102
  eval_steps_per_second   =      3.322
```

(ii)

+ Code + Markdown

```
.6]:
batch_size=6
args=TrainingArguments(
    output_dir='output_dir',
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=3,
    evaluation_strategy = "epoch",
    learning_rate=4e-5
)
```

[5001/5001 1:01:11, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Loc Precision	Loc Recall	Loc F1	Loc Number	Org Precision	Org Recall	Org F1	Org Number	Per Precision	Per Recall	Per F1	Per Number	Overall Precision	Overall Recall	Overall F1	Overall Accuracy
1	0.303600	0.314761	0.716996	0.614217	0.661639	10213	0.611366	0.353975	0.448356	9786	0.686152	0.593111	0.636248	10568	0.679618	0.523604	0.591496	0.905525
2	0.242100	0.264849	0.743808	0.676295	0.708447	10213	0.546502	0.547619	0.547060	9786	0.715644	0.683952	0.699439	10568	0.667786	0.637747	0.652421	0.917833
3	0.199100	0.262641	0.718279	0.727210	0.722717	10213	0.566479	0.556407	0.561398	9786	0.716231	0.696915	0.706441	10568	0.669324	0.662054	0.665669	0.920160

(iii)

Epoch	Training Loss	Validation Loss	Loc Precision	Loc Recall	Loc F1	Loc Number	Org Precision	Org Recall	Org F1	Org Number	Per Precision	Per Recall	Per F1	Per Number	Overall Precision	Overall Recall	Overall F1	Overall Accuracy
1	0.471800	0.441183	0.497876	0.436111	0.464951	10213	0.318898	0.198651	0.244805	9786	0.510353	0.501419	0.505847	10568	0.460059	0.382668	0.417810	0.870880
2	0.385000	0.388195	0.565406	0.485851	0.522618	10213	0.393599	0.354384	0.372963	9786	0.609916	0.517979	0.560201	10568	0.523455	0.454870	0.486758	0.884538
3	0.361400	0.377335	0.550163	0.543915	0.547021	10213	0.417843	0.357041	0.385056	9786	0.616870	0.532173	0.571400	10568	0.532093	0.480027	0.504721	0.887121

```
args=TrainingArguments(
    output_dir='output_dir',
    per_device_train_batch_size=10,
    per_device_eval_batch_size=10,
    num_train_epochs=3,
    evaluation_strategy = "epoch",
    learning_rate=5e-6)
```

Indic-NER -

(i)

Epoch	Training Loss	Validation Loss	Loc Precision	Loc Recall	Loc F1	Loc Number	Org Precision	Org Recall	Org F1	Org Number	Per Precision	Per Recall	Per F1	N
1	0.153100	0.170914	0.810615	0.855380	0.832396	10213	0.683332	0.677396	0.680351	9786	0.806352	0.838475	0.822100	
2	0.103200	0.182538	0.817916	0.850191	0.833741	10213	0.673960	0.698753	0.686133	9786	0.804627	0.829296	0.816775	
3	0.074200	0.207841	0.811579	0.852345	0.831463	10213	0.671473	0.685265	0.678299	9786	0.800625	0.824186	0.812235	

```
batch_size=8
args=TrainingArguments(
    output_dir='output_dir',
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=3,
    evaluation_strategy = "epoch",
    learning_rate=4e-5)
```

(ii)

```
batch_size=10
args=TrainingArguments(
    output_dir='output_dir',
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=3,
    evaluation_strategy = "epoch",
    learning_rate=4e-5)
```

Loc F1	Loc Number	Org Precision	Org Recall	Org F1	Org Number	Per Precision	Per Recall	Per F1	Per Number	Overall Precision	Overall Recall	Overall F1	Overall Accuracy
0.822500	10213	0.664523	0.686389	0.675279	9786	0.796529	0.820874	0.808519	10568	0.758121	0.783394	0.770550	0.944361
0.820845	10213	0.659564	0.676885	0.668112	9786	0.788212	0.821253	0.804393	10568	0.754686	0.778388	0.766354	0.943884
0.817450	10213	0.648524	0.675761	0.661863	9786	0.788395	0.815102	0.801526	10568	0.748886	0.775542	0.761981	0.943076

(iii)

```
batch_size=16
args=TrainingArguments(
    output_dir='output_dir',
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=3,
    evaluation_strategy = "epoch",
    learning_rate=5e-5)
```

[1875/1875 1:08:08, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Loc Precision	Loc Recall	Loc F1	Loc Number	Org Precision	Org Recall	Org F1	Org Number	Per Precision	Per Recall	Per F1	Per Number	Overall Precision	Overall Recall	Overall F1	Overall Accuracy
1	6.928800	2.577833	0.000169	0.000783	0.000278	10213	0.016926	0.011241	0.013509	9786	0.124781	0.208460	0.156114	10568	0.032456	0.075932	0.045474	0.076111
2	1.635000	0.648141	0.000209	0.000098	0.000133	10213	0.017225	0.005722	0.008591	9786	0.080736	0.068887	0.074343	10568	0.046025	0.025681	0.032967	0.827768
3	0.655200	0.585783	0.000164	0.000098	0.000123	10213	0.024151	0.009810	0.013952	9786	0.101530	0.118660	0.109429	10568	0.060286	0.044198	0.051003	0.830676

CONCLUSION -

From the above results, we can conclude that the larger batch-size overfits the model and accuracy goes down whereas smaller batch size affects the accuracy but not as drastically as increasing it.Learning rate too plays a significant role in the accuracy as optimal learning rate helps increase the accuracy of the model on general data.