Comparison of ChatGPT, IndicBERT, and IndicNER against Ground truth:

The evaluation metrics used include precision, recall, F1-score, support, accuracy, macro F1-score, micro F1-score, and weighted average F1-score.

ChatGPT Output:

	precision	recall	f1-score	support
B-LOC	1.00	0.80	0.89	5
B-MISC	0.00	0.00	0.00	14
B-ORG	0.00	0.00	0.00	5
B-PER	1.00	0.64	0.78	11
I-MISC	0.00	0.00	0.00	7
I-ORG	0.00	0.00	0.00	2
I-PER	1.00	1.00	1.00	7
O	0.95	1.00	0.97	634
accuracy			0.95	685
macro avg	0.49	0.43	0.46	685
weighted avg	0.91	0.95	0.93	685

Macro F1 Score: 0.40458906254448146

IndicBERT Output:

	precision	recall	f1-score	support
I-PER	0.50	0.43	0.46	7
B-LOC	0.50	0.40	0.44	5
I-ORG	0.25	0.50	0.33	2
B-ORG	0.33	0.40	0.36	5
I-MISC	0.00	0.00	0.00	7
B-PER	0.44	0.36	0.40	11
O	0.95	0.94	0.94	634
B-MISC	0.00	0.00	0.00	14
micro avg	0.92	0.89	0.91	685
macro avg	0.37	0.38	0.37	685
weighted avg	0.90	0.89	0.89	685

Macro F1 Score: 0.3274152491888042 7

IndicNER Output:

	precision	recall	f1-score	support
I-PER	0.50	0.43	0.46	7
B-LOC	0.25	0.20	0.22	5
I-ORG	0.14	0.50	0.22	2
B-ORG	0.25	0.40	0.31	5
I-MISC	0.00	0.00	0.00	7
B-PER	0.38	0.45	0.42	11
O	0.96	0.95	0.95	634
B-MISC	0.00	0.00	0.00	14
micro avg	0.92	0.89	0.91	685
macro avg	0.31	0.37	0.32	685
weighted avg	0.90	0.89	0.90	685

Macro F1 Score: 0.2868015014079462

1. **Overall Performance:** ChatGPT outperforms both IndicBERT and IndicNER in terms of overall accuracy and F1-score, achieving an accuracy of 95% and a weighted average F1-score of 93%. IndicBERT and IndicNER have similar performance metrics, with micro F1-scores of 91% and 91%, respectively, and weighted average F1-scores of 89%.

2. **Class-wise Performance:** ChatGPT shows high precision and recall for B-LOC and B-PER entities, with perfect precision and recall for B-PER entities, and very high recall for B-LOC entities (80%). However, ChatGPT shows poor performance for B-MISC and B-ORG entities, with precision and recall scores of 0% for both classes. IndicBERT and IndicNER show similar patterns of performance across different entity types, with relatively low precision, recall, and F1-scores for all classes compared to ChatGPT.

3. **Comparison between IndicBERT and IndicNER:** IndicBERT generally performs slightly better than IndicNER across all metrics. Both IndicBERT and IndicNER show poor performance for B-MISC and I-MISC entities, with precision, recall, and F1-scores close to 0. IndicBERT and IndicNER both have difficulties in recognizing certain entity types, especially those with less training data or less distinct patterns.

4. **Macro F1-score:** ChatGPT achieves a much higher macro F1-score compared to both IndicBERT and IndicNER, indicating better overall performance across all classes. IndicBERT and IndicNER have lower macro F1-scores, suggesting that their performance is more inconsistent across different classes.

5. **Interpretation:** ChatGPT demonstrates superior performance in named entity recognition compared to IndicBERT and IndicNER, especially for certain entity types like B-LOC and B-PER. IndicBERT and IndicNER show weaknesses in recognizing certain entity types, especially B-MISC and I-MISC, indicating a need for improvement in their training data or model architecture. In summary, ChatGPT exhibits the best performance in named entity recognition among the evaluated systems, while IndicBERT and IndicNER show weaker performance, especially for certain entity types. Further optimization and fine-tuning of these models may help improve their performance in identifying named entities accurately.

Hyperparameters and Their Significance

Number of Training Epochs (trainer.args.num_train_epochs):

- Significance: This hyperparameter defines the number of times the entire training dataset is passed through the model.
- Optimal Value: 3 epochs.
- Justification: Assignment requirement.

Evaluation Strategy (trainer.args.evaluation_strategy):

- Significance: Specifies when to evaluate the model during training, either at the end of each epoch or after a certain number of training steps.
- Optimal Value: "epoch" (evaluate at the end of each epoch).
- Justification: Assignment requirement.

Evaluation Steps (trainer.args.eval_steps):

- Significance: Determines the frequency of model evaluation during training when the evaluation strategy is set to steps.
- Optimal Value: 500 steps.
- Justification: Evaluating the model every 500 steps balances the computational cost of frequent evaluations with the need for timely feedback on the model's performance and convergence.

Learning Rate (trainer.args.learning_rate):

- Significance: Controls the step size during the optimization process, influencing the magnitude of parameter updates.
- Optimal Value: 3e-5.
- Justification: 3e-5 is a commonly used learning rate for fine-tuning pre-trained language models. It allows for stable convergence without the risk of overshooting the optimal parameter values.

Per-device Training Batch Size (trainer.args.per_device_train_batch_size):

- Significance: Specifies the number of training samples processed simultaneously on each device (e.g., GPU or TPU).
- Optimal Value: 16.
- Justification: A batch size of 16 strikes a balance between training efficiency and memory constraints. It allows for efficient GPU utilization while avoiding excessive memory consumption.

Weight Decay (trainer.args.weight_decay):

- Significance: Regularizes the model by penalizing large weights during optimization, helping prevent overfitting.
- Optimal Value: 0.01.
- Justification: A weight decay of 0.01 provides effective regularization without excessively dampening the learning process, thus helping the model generalize better to unseen data.

Gradient Accumulation Steps (trainer.args.gradient_accumulation_steps):

- Significance: Accumulates gradients over multiple steps before updating model parameters, effectively increasing the effective batch size.
- Optimal Value: 1 (no accumulation).
- Justification: In this case, no gradient accumulation is performed. Each batch of size 16 is processed independently to update model parameters, ensuring stable and consistent optimization.

Warm-up Steps (trainer.args.warmup_steps):

- Significance: Gradually increases the learning rate during the initial training steps, allowing the model to explore the parameter space more effectively.
- Optimal Value: 300 steps.
- Justification: 300 warm-up steps provide a sufficient period for the learning rate to ramp up gradually, helping the model avoid getting stuck in suboptimal parameter configurations early in training.

Output of IndicBERT:

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	0.624600	0.340406	0.568740	0.680505	0.619623	10213	0.524615	0.389843	0.447297	9786	0.606827	0.635882	0.621015	10568	0.572041	0.572022	0.572031	0.897575
2	0.311000	0.291150	0.700156	0.659845	0.679403	10213	0.526807	0.508073	0.517270	9786	0.703289	0.641465	0.670956	10568	0.644206	0.604901	0.623935	0.911345
3	0.266000	0.284142	0.687506	0.690199	0.688850	10213	0.538144	0.519722	0.528773	9786	0.698874	0.663891	0.680934	10568	0.643883	0.626525	0.635085	0.912943

Eval matrix of IndicBERT:

```
epoch                   =        3.0
eval_LOC_f1             =     0.6888
eval_LOC_number         =      10213
eval_LOC_precision      =     0.6875
eval_LOC_recall         =     0.6902
eval_ORG_f1             =     0.5288
eval_ORG_number         =       9786
eval_ORG_precision      =     0.5381
eval_ORG_recall         =     0.5197
eval_PER_f1             =     0.6809
eval_PER_number         =      10568
eval_PER_precision      =     0.6989
eval_PER_recall         =     0.6639
eval_loss               =     0.2841
eval_overall_accuracy   =     0.9129
eval_overall_f1         =     0.6351
eval_overall_precision  =     0.6439
eval_overall_recall     =     0.6265
eval_runtime            = 0:04:13.01
eval_samples_per_second =     53.197
eval_steps_per_second   =      3.328
```

Train matrix of IndicBERT:

```
eval_LOC_f1                =     0.7352
eval_LOC_number            =      14841
eval_LOC_precision         =     0.7335
eval_LOC_recall            =     0.7369
eval_ORG_f1                =     0.5876
eval_ORG_number            =      14082
eval_ORG_precision         =     0.6003
eval_ORG_recall            =     0.5754
eval_PER_f1                =     0.7522
eval_PER_number            =      15614
eval_PER_precision         =     0.7715
eval_PER_recall            =     0.7339
eval_loss                  =      0.232
eval_overall_accuracy      =     0.9313
eval_overall_f1            =     0.6947
eval_overall_precision     =      0.705
eval_overall_recall        =     0.6848
eval_runtime               = 0:06:13.94
eval_samples_per_second    =     53.484
eval_steps_per_second      =      3.343
```

Output of IndicNER:

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	1.374100	0.175191	0.800273	0.861549	0.829781	10213	0.690043	0.689761	0.689902	9786	0.809954	0.839232	0.824333	10568	0.769628	0.798835	0.783960	0.947742
2	0.133600	0.174975	0.817909	0.854108	0.835616	10213	0.684968	0.690987	0.687964	9786	0.801904	0.836961	0.819057	10568	0.770742	0.795956	0.783146	0.947890
3	0.107100	0.187496	0.811127	0.856555	0.833222	10213	0.673400	0.691498	0.682329	9786	0.803743	0.833176	0.818194	10568	0.765045	0.795629	0.780037	0.946911

Eval matrix of IndicNER:

```
epoch                       =             3.0
eval_LOC_f1                 =          0.8332
eval_LOC_number             =           10213
eval_LOC_precision          =          0.8111
eval_LOC_recall             =          0.8566
eval_ORG_f1                 =          0.6823
eval_ORG_number             =            9786
eval_ORG_precision          =          0.6734
eval_ORG_recall             =          0.6915
eval_PER_f1                 =          0.8182
eval_PER_number             =           10568
eval_PER_precision          =          0.8037
eval_PER_recall             =          0.8332
eval_loss                   =          0.1875
eval_overall_accuracy       =          0.9469
eval_overall_f1             =            0.78
eval_overall_precision      =           0.765
eval_overall_recall         =          0.7956
eval_runtime                = 0:05:11.78
eval_samples_per_second     =           43.17
eval_steps_per_second       =           2.701
```

Train matrix of IndicNER:

```
  eval_LOC_f1                =     0.8999
  eval_LOC_number            =      14841
  eval_LOC_precision         =     0.8814
  eval_LOC_recall            =     0.9193
  eval_ORG_f1                =     0.8212
  eval_ORG_number            =      14082
  eval_ORG_precision         =     0.8164
  eval_ORG_recall            =     0.8262
  eval_PER_f1                =     0.8993
  eval_PER_number            =      15614
  eval_PER_precision         =     0.8906
  eval_PER_recall            =     0.9081
  eval_loss                  =     0.0731
  eval_overall_accuracy      =     0.9762
  eval_overall_f1            =      0.875
  eval_overall_precision     =     0.8643
  eval_overall_recall        =     0.8859
  eval_runtime               = 0:07:42.61
  eval_samples_per_second    =     43.233
  eval_steps_per_second      =      2.702
```

Comparison of IndicBERT and IndicNER Performance

1. **Overall Performance Metrics:**
    - IndicBERT: Achieves an overall F1-score of 0.6351 and an overall accuracy of 0.9129 after 3 epochs of training.
    - IndicNER: Performs slightly better with an overall F1-score of 0.7800 and an overall accuracy of 0.9469 after 3 epochs.

2. **Entity-wise Performance:**
    - IndicBERT: Shows relatively balanced performance across different entity types, with F1-scores ranging from 0.5288 (ORG) to 0.6809 (PER).
    - IndicNER: Demonstrates superior performance across all entity types compared to IndicBERT, with F1-scores ranging from 0.6823 (ORG) to 0.8332 (LOC).

3. **Training and Evaluation Metrics:**
    - IndicBERT:
        - Achieves higher F1-scores during training compared to evaluation, indicating some degree of overfitting.
        - Training F1-scores: LOC (0.7352), ORG (0.5876), PER (0.7522), Overall (0.6947).
    - IndicNER:
        - Shows consistent performance between training and evaluation phases, suggesting better generalization.
        - Training F1-scores: LOC (0.8999), ORG (0.8212), PER (0.8993), Overall (0.875).
        -

4. **Runtime and Efficiency:**
    - IndicBERT: Evaluation runtime is shorter compared to training, with 3.328 steps per second during evaluation.
    - IndicNER: Evaluation runtime is also shorter compared to training, with 2.701 steps per second during evaluation.

5. **Key Insights:**
    - IndicNER outperforms IndicBERT in terms of overall performance and entity-wise F1-scores, indicating its effectiveness in capturing named entities in Indic languages.
    - IndicBERT shows signs of overfitting as evidenced by the disparity between training and evaluation F1-scores.
    - IndicNER demonstrates better generalization, with consistent performance across training and evaluation phases.
    - Both models achieve high overall accuracy, suggesting their effectiveness in accurately recognizing named entities.

Conclusion:
IndicNER emerges as the superior model for named entity recognition in Indic languages, showcasing better generalization and performance across various entity types. Its consistent performance between training and evaluation phases indicates robustness and reliability.