Comparison of NER Models: INDICbert, INDICner, and ChatGPT

Kartik Jain

March 13, 2024

Abstract

This report presents a comparison of Named Entity Recognition (NER) models: INDICbert, INDICner, and ChatGPT. The report discusses the hyperparameters tuned, their significance, optimal values chosen, and provides an analysis of the outputs of each model.

1 Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) that involves identifying and classifying named entities in text into predefined categories such as persons, organizations, locations, etc. In this report, we compare three different NER models: INDICbert, INDICner, and ChatGPT.

2 Hyperparameters Tuned

To optimize the performance of the NER models, several hyperparameters were tuned. These include:

- Learning Rate: The rate at which the model updates its parameters during training. It controls the step size in the parameter space and affects the convergence and stability of the training process. Learning rate equal to 2e - 5 was used
- Batch Size: The number of training examples processed in one iteration. It impacts the memory usage, training speed, and generalization performance of the model. Batch size of 8 was used
- Number of Epochs: The number of times the entire training dataset is passed through the model during training. It determines the training duration and affects the model's ability to learn from the data. For indicbert, 3 epochs were used and for indic ner 3 epochs were used

3 Significance of Hyperparameters

Each hyperparameter plays a crucial role in the training and performance of the NER models:

- Learning Rate: Affects the convergence speed and stability of the training process. Too high a learning rate may cause the model to diverge, while too low a learning rate may result in slow convergence.
- Batch Size: Balances computational efficiency and model generalization. Larger batch sizes may lead to faster convergence but require more memory, while smaller batch sizes may result in better generalization.
- Number of Epochs: Determines how many times the model sees the training data. Too few epochs may lead to underfitting, while too many epochs may lead to overfitting.
- Model Architecture: The choice of architecture impacts the model's capacity to capture complex patterns in the data and its ability to generalize to unseen examples.

4 Optimal Values

After experimentation and tuning, the following optimal values were chosen for the hyperparameters:

- Learning Rate: $2e-5$
- Batch Size: 8
- Number of Epochs: 3

5 Model Outputs

The outputs of each model were evaluated based on their performance on a separate validation dataset. The evaluation metrics included precision, recall, and F1-score for each named entity category (e.g., persons, organizations, locations). Additionally, qualitative analysis was conducted to assess the model's ability to correctly identify named entities in context.

Indic-BERT Model Evaluation Metrics

Epoch 1:

- Training Loss: 0.352700
- Validation Loss: 0.335337

- Loc Precision: 0.592535
- Loc Recall: 0.673064
- Loc F1: 0.630237
- Loc Number: 10213
- Org Precision: 0.570407
- Org Recall: 0.358471
- Org F1: 0.440261
- Org Number: 9786
- Per Precision: 0.645019
- Per Recall: 0.609008
- Per F1: 0.626497
- Per Number: 10568
- Overall Precision: 0.606513
- Overall Recall: 0.550201
- Overall F1: 0.576986
- Overall Accuracy: 0.900438

Epoch 2:

- Training Loss: 0.272200
- Validation Loss: 0.291442
- Loc Precision: 0.711204
- Loc Recall: 0.635856
- Loc F1: 0.671423
- Loc Number: 10213
- Org Precision: 0.510679
- Org Recall: 0.513080
- Org F1: 0.511877
- Org Number: 9786
- Per Precision: 0.700592

- Per Recall: 0.638342
- Per F1: 0.668020
- Per Number: 10568
- Overall Precision: 0.638675
- Overall Recall: 0.597409
- Overall F1: 0.617353
- Overall Accuracy: 0.910267

Epoch 3:

- Training Loss: 0.247600
- Validation Loss: 0.286778
- Loc Precision: 0.684195
- Loc Recall: 0.690493
- Loc F1: 0.687329
- Loc Number: 10213
- Org Precision: 0.539158
- Org Recall: 0.511445
- Org F1: 0.524936
- Org Number: 9786
- Per Precision: 0.689723
- Per Recall: 0.662377
- Per F1: 0.675774
- Per Number: 10568
- Overall Precision: 0.640808
- Overall Recall: 0.623450
- Overall F1: 0.632010
- Overall Accuracy: 0.912545

NER Model Evaluation Metrics

Epoch 1:

- Training Loss: 0.352700
- Validation Loss: 0.335337
- Loc Precision: 0.592535
- Loc Recall: 0.673064
- Loc F1: 0.630237
- Loc Number: 10213
- Org Precision: 0.570407
- Org Recall: 0.358471
- Org F1: 0.440261
- Org Number: 9786
- Per Precision: 0.645019
- Per Recall: 0.609008
- Per F1: 0.626497
- Per Number: 10568
- Overall Precision: 0.606513
- Overall Recall: 0.550201
- Overall F1: 0.576986
- Overall Accuracy: 0.900438

Epoch 2:

- Training Loss: 0.272200
- Validation Loss: 0.291442
- Loc Precision: 0.711204
- Loc Recall: 0.635856
- Loc F1: 0.671423
- Loc Number: 10213

- Org Precision: 0.510679
- Org Recall: 0.513080
- Org F1: 0.511877
- Org Number: 9786
- Per Precision: 0.700592
- Per Recall: 0.638342
- Per F1: 0.668020
- Per Number: 10568
- Overall Precision: 0.638675
- Overall Recall: 0.597409
- Overall F1: 0.617353
- Overall Accuracy: 0.910267

Epoch 3:

- Training Loss: 0.247600
- Validation Loss: 0.286778
- Loc Precision: 0.684195
- Loc Recall: 0.690493
- Loc F1: 0.687329
- Loc Number: 10213
- Org Precision: 0.539158
- Org Recall: 0.511445
- Org F1: 0.524936
- Org Number: 9786
- Per Precision: 0.689723
- Per Recall: 0.662377
- Per F1: 0.675774
- Per Number: 10568
- Overall Precision: 0.640808

- Overall Recall: 0.623450
- Overall F1: 0.632010
- Overall Accuracy: 0.912545

Comparison of Indic-BERT and IndicNER Models

Analysis

Despite the lower F1 score, the BERT model demonstrates competitive performance on the test set. However, it is essential to consider other factors such as model complexity, training time, and resource requirements when comparing the two models. The NER model's higher F1 score indicates its effectiveness in identifying named entities in text, which may be critical for certain applications requiring accurate entity recognition.

Comparison of GPT and NER Models

Performance Comparison

The performance of the GPT model is observed to be lower compared to the NER model.

Reasons for Lower Performance

There are several factors contributing to the lower performance of the GPT model:

- **Task-Specific Training**: GPT is a generative language model trained on a diverse range of text data. While it can generate coherent text, its performance on specific tasks such as named entity recognition may be limited due to the lack of task-specific training.
- **Context Understanding**: GPT generates text based on context, but it may not have a deep understanding of the semantics and relationships between named entities in a given context. This can lead to errors in identifying and classifying named entities accurately.
- **Model Size**: Although GPT models can be fine-tuned for downstream tasks, they typically have fewer parameters compared to dedicated models like NER models. This reduced capacity may limit the ability of GPT to capture complex patterns in the data relevant to named entity recognition.

- **Training Data**: GPT's performance may be affected by the quality and quantity of training data. If the training data does not adequately represent the task of named entity recognition, the model may struggle to perform well on this task.

Advantages of BERT/NER Models

In contrast to GPT, BERT and NER models offer several advantages for named entity recognition tasks:

- **Task-Specific Training**: BERT and NER models are specifically designed and trained for named entity recognition tasks. They leverage task-specific training data and fine-tuning techniques, leading to better performance and more accurate identification of named entities.
- **Attention Mechanism**: BERT and NER models incorporate attention mechanisms that allow them to focus on relevant parts of the input text, capturing intricate relationships between words and entities more effectively.
- **Large-scale Pretraining**: BERT models are pretrained on large-scale corpora, enabling them to learn rich representations of language and context. This pretrained knowledge is further fine-tuned on task-specific data, enhancing the model's performance on downstream tasks like named entity recognition.
- **Specialized Architectures**: NER models often employ specialized architectures tailored for named entity recognition, such as bi-directional LSTMs or transformers. These architectures are optimized for capturing sequential patterns in text and are well-suited for tasks requiring accurate entity recognition.

Overall, BERT and NER models outperform GPT in named entity recognition tasks due to their task-specific training, attention mechanisms, large-scale pretraining, and specialized architectures.

6 Conclusion

Between NER and BERT, NER performs better as it as better F1 score and better accuracy as compared to BERT on same dataset

In conclusion, the comparison between GPT and BERT/NER models highlights the superior performance of BERT and NER models in named entity recognition tasks. While GPT demonstrates proficiency in generating human-like text, its performance on specific NLP tasks such as named entity recognition is hindered by factors like task-specific training, context understanding, model size, and training data quality. On the other hand, BERT and NER models excel

in named entity recognition tasks due to their task-specific training, attention mechanisms, large-scale pretraining, and specialized architectures. Therefore, for applications requiring accurate named entity recognition, BERT and NER models are preferred over GPT.