Named Entity Recognition using BERT and BiLSTM Models

Youssef Mohamed

July 2024

Abstract

This report presents a comprehensive analysis of Named Entity Recognition (NER) using two distinct models: a Bidirectional Long Short-Term Memory (BiLSTM) network and a Bidirectional Encoder Representations from Transformers (BERT) model. The objective is to compare their performance in terms of accuracy and model size. Both models achieved an impressive accuracy of 97%, with the BiLSTM model being significantly smaller in size (0.6 MB) compared to the BERT model (268 MB).

1 Introduction

Named Entity Recognition (NER) is a crucial task in Natural Language Processing (NLP) that involves identifying and classifying entities such as names of people, organizations, locations, and other proper nouns in text. This report explores two advanced techniques for NER: BiLSTM and BERT. The BiLSTM model leverages the capabilities of Recurrent Neural Networks (RNNs), while the BERT model utilizes transformer architecture for enhanced contextual understanding.

2 Data Description

The dataset [4] used in this study comprises sentences annotated with Part-of-Speech (POS) tags and Named Entity Recognition (NER) tags. Each sentence in the dataset is associated with its corresponding POS and NER tags, which are essential for the training and evaluation of the models.

Sentence #	Sentence
5957	He said Mr. Blair did not have a lawyer present and was treated as a witness rather than a suspect.
22365	Meanwhile, insurgent attacks continue in Iraq.
9132	Caracas airport official Jose Cabello denies this, saying the group made no attempt to contact Venezuelan authorities.

Table 1: Examples of Sentences in the Dataset

Sentence #	POS Tags
5957	['PRP', 'VBD', 'NNP', 'NNP', 'VBD', 'RB', 'VB', 'DT', 'NN', 'NN', 'CC', 'VBD', 'VBN', 'IN', 'DT', 'NN', 'RB', 'IN', 'DT', 'NN', '.']
22365	['RB', ',', 'JJ', 'NNS', 'VBP', 'IN', 'NNP', '.']
9132	['NNP', 'NN', 'NN', 'NNP', 'NNP', 'VBZ', 'DT', ',', 'VBG', 'DT', 'NN', 'VBD', 'DT', 'NN', 'TO', 'VB', 'JJ', 'NNS', '.']

Table 2: Examples of POS Tags Corresponding to the Sentences

Sentence #	NER Tags
5957	['O', 'O', 'B-per', 'I-per', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
22365	['O', 'O', 'O', 'O', 'O', 'O', 'B-geo', 'O']
9132	['B-geo', 'O', 'O', 'B-per', 'I-per', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'B-gpe', 'O', 'O']

Table 3: Examples of NER Tags Corresponding to the Sentences

The dataset was split into training and testing sets with an 80-20 split, resulting in 38,367 training samples and 9,592 testing samples.

3 Data Preprocessing

Data preprocessing is a crucial step to ensure that the input data is in the correct format for the models. The preprocessing steps involved the following:

- **Conversion of Tags:** The POS and NER tags, originally in string format, were converted to list format to facilitate processing. For example:
  - Sentence: "He said Mr. Blair did not have a lawyer present and was treated as a witness rather than a suspect."
  - POS Tags: ['PRP', 'VBD', 'NNP', 'NNP', 'VBD', 'RB', 'VB', 'DT', 'NN', 'NN', 'CC', 'VBD', 'VBN', 'IN', 'DT', 'NN', 'RB', 'IN', 'DT', 'NN', '.']
  - NER Tags: ['O', 'O', 'B-per', 'I-per', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
- **Tokenization:** Sentences and tags were tokenized to convert them into sequences of integers. This step is necessary for embedding layers to process the text data.
- **Padding:** Sequences were padded to ensure uniform input size for the models. Padding is crucial for handling variable-length sequences in batch processing.

These preprocessing steps ensured that the data was properly formatted for input into the BiLSTM and BERT models.

4 Baseline Experiments: BiLSTM Model

The BiLSTM model architecture includes an embedding layer, a masking layer, a bidirectional LSTM layer, and a dense layer. The embedding layer maps input words to dense vectors of fixed size, the masking layer handles variable-length sequences, the bidirectional LSTM captures dependencies in both forward and backward directions, and the dense layer outputs the NER tags.

4.1 Model Architecture

The BiLSTM model can be mathematically represented as follows:

$$\mathbf{x}_t = \text{Embedding}(\mathbf{w}_t) \tag{1}$$

$$\mathbf{h}_t = \text{BiLSTM}(\mathbf{x}_t) \tag{2}$$

$$\mathbf{y}_t = \text{Dense}(\mathbf{h}_t) \tag{3}$$

where $\mathbf{w}_t$ is the input word at time step t, $\mathbf{x}_t$ is the corresponding embedding, $\mathbf{h}_t$ is the hidden state from the BiLSTM layer, and $\mathbf{y}_t$ is the output NER tag.

4.2 Training and Evaluation

The model was trained using the Adam optimizer and Sparse Categorical Crossentropy loss function. Early stopping and model checkpointing were employed to prevent overfitting. The evaluation metrics included accuracy, precision, recall, and F1-score.

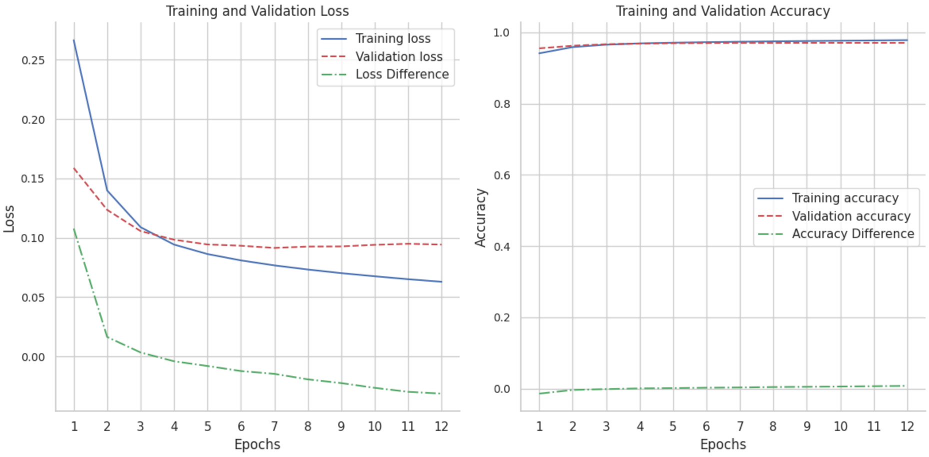

Figure 1: BiLSTM accuracy-Loss report.

5 Advanced Experiments: BERT Model

The BERT model was implemented using the Hugging Face Transformers library. The DistilBERT variant was chosen for its balance between performance and computational efficiency.

5.1 Model Architecture

BERT's architecture is based on the Transformer model, which uses self-attention mechanisms to capture context from both directions. The key components include the multi-head self-attention mechanism and the position-wise feed-forward network.

The self-attention mechanism can be described as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

where Q (queries), K (keys), and V (values) are projections of the input embeddings, and d_k is the dimensionality of the keys.

5.2 Training and Evaluation

The BERT model was fine-tuned on the NER task using a custom training loop with evaluation metrics including precision, recall, F1-score, and accuracy. The training process was monitored using the `Trainer` class from the Hugging Face library.

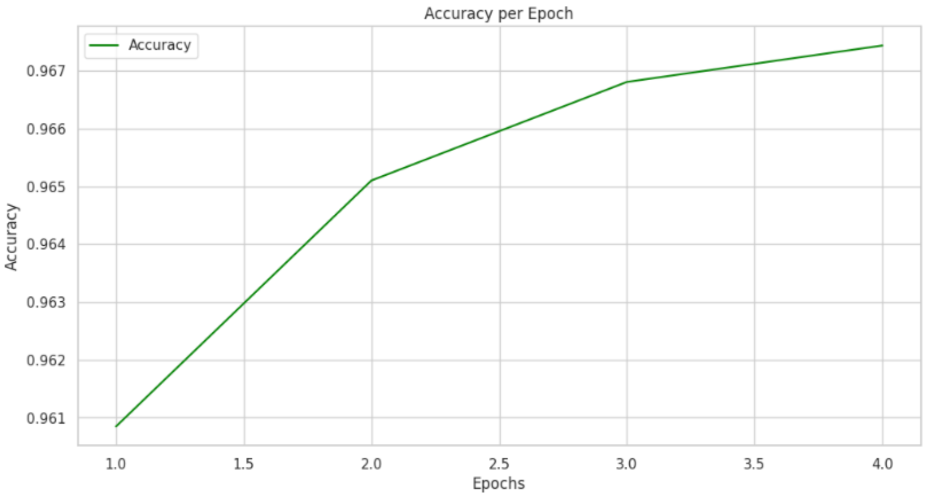

Figure 2: distilBERT accuracy per epoch.

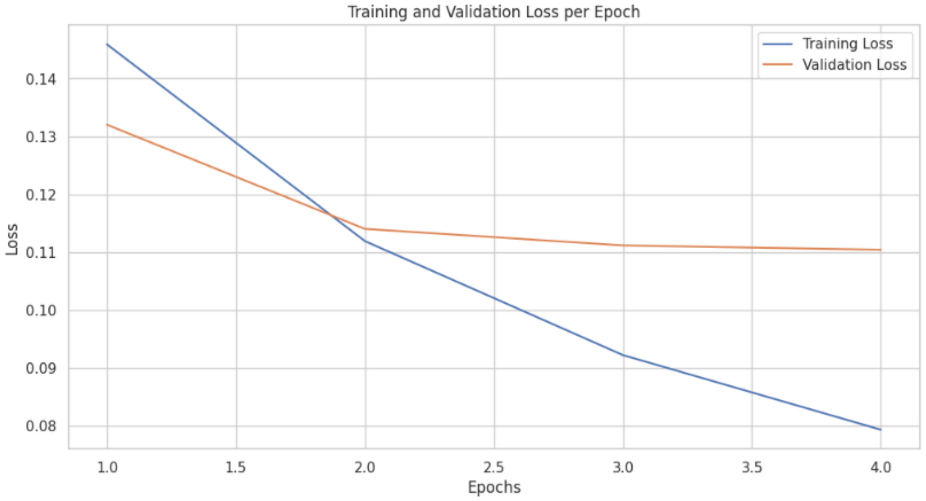

Figure 3: distilBERT loss per epoch.

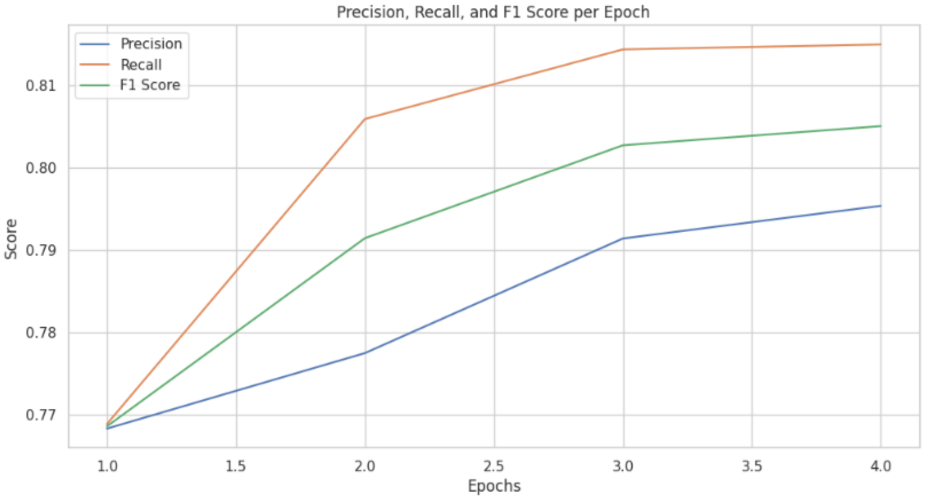

Figure 4: distilBERT scores per epoch.

6 Evaluation Metrics

In the evaluation of Named Entity Recognition (NER) models, several metrics are employed to assess the model's performance. This section provides a detailed explanation of the metrics used in this study and the rationale behind their selection.

6.1 Precision

Precision is a measure of the accuracy of the predicted entities. It is defined as the ratio of true positive predictions to the sum of true positives and false positives. Mathematically, it is expressed as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

where TP represents the number of true positives, and FP denotes the number of false positives. Precision is crucial in NER tasks as it reflects how many of the predicted entities are actually correct. High precision indicates that the model makes fewer false positive predictions.

6.2 Recall

Recall measures the ability of the model to identify all relevant entities. It is defined as the ratio of true positive predictions to the sum of true positives and false negatives. Mathematically, it is expressed as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

where FN represents the number of false negatives. Recall is essential in NER as it indicates how many of the actual entities were successfully identified by the model. High recall means the model successfully captures most of the relevant entities.

6.3 F1 Score

The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both aspects. It is defined as:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

The F1 Score is particularly useful in NER tasks where both precision and recall are important. It ensures that the model's performance is evaluated comprehensively by taking into account both false positives and false negatives.

6.4 Accuracy

Accuracy is the ratio of the number of correct predictions (both true positives and true negatives) to the total number of predictions. It is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

where TN denotes the number of true negatives. While accuracy is a straightforward measure, it may not always be the most informative metric in imbalanced datasets where some classes are much more frequent than others. However, it provides an overall view of the model's correctness.

6.5 Why These Metrics?

The selection of precision, recall, F1 Score, and accuracy is based on the nature of the NER task and the requirements for model evaluation:

- **Precision** and **Recall** are fundamental for evaluating how well a model identifies relevant entities while avoiding false positives and false negatives. In NER, both false positives (incorrectly tagged entities) and false negatives (missed entities) have significant implications.

- **F1 Score** combines precision and recall into a single metric, providing a balanced view of the model's performance. It is especially useful when there is a need to balance the trade-off between precision and recall.

- **Accuracy** gives an overall measure of correctness but should be considered alongside other metrics to ensure a comprehensive evaluation, particularly in cases of class imbalance.

7 Results and Discussion

Both models achieved a remarkable accuracy of 97%. However, the BiLSTM model was significantly more compact at 0.6 MB compared to the BERT model's 268 MB. This substantial difference in model size has implications for deployment and resource utilization. The evaluation metrics for both models are summarized in Table 4.

Metric	**BiLSTM Model**	**BERT Model**
Accuracy	97%	97%
Model Size	0.6 MB	268 MB

Table 4: Comparison of BiLSTM and BERT Models

8 Conclusion

This study demonstrates that both BiLSTM and BERT models are highly effective for NER tasks, achieving high accuracy and robust performance metrics. However, the choice between these models should consider the trade-off between model size and performance. The BiLSTM model is preferable for resource-constrained environments, whereas the BERT model offers superior contextual understanding at the cost of increased computational resources.

References

[1] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805.*

[2] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. *arXiv preprint arXiv:1706.03762.*

[4] Naser Alqaydeh, *Named Entity Recognition (NER) Corpus*, Kaggle, 2021. https://www.kaggle.com/datasets/naseralqaydeh/named-entity-recognition-ner-corpus. Accessed: 2024-07-20.