Token-Level Multi-Class Classification of POS Tags using Deep Learning Techniques

Abstract

This project aims to use various deep learning techniques to perform token-level multi-class classification for Parts-of-Speech (POS) tagging. The deep learning models used were Recurrent Neural Networks (RNN) [1], Gated Recurrent Units (GRU) [4], Long Short-Term Memory networks (LSTM) [3], and Bidirectional LSTM (BiLSTM) [5] to evaluate their performance on POS classification, on a predetermined dataset. Tuning of hyperparameters such as learning rate, number of units, batch size and epochs led to variations in performance metrics for each model, which were analysed. After model evaluation, it was found that BiLSTM emerged as the best-performing architecture with F-1 scores surpassing the rest.

Introduction

POS tagging is fundamental to Natural Language Processing (NLP) for syntactic analysis and translation. The aim behind this study is to observe and compare the performance of various deep learning models to discover the most accurate and efficient model for the task of POS tagging. The dataset is preprocessed to token level for modularity and detailed training, and as POS tags are mostly assigned at word level.

Methodology

1. Data Exploration and Preprocessing

- **Dataset:** A structured dataset split into training (80%) and testing (20%) sets were done and used. An additional 10% was later split from the training set as the cross validation set to compare in between epochs.
- **Exploratory Data Analysis (EDA):** Evaluated sentence length distribution and unique POS tag frequency.
- **Tokenization:** Converted text into numerical sequences using a vocabulary index.
- **Padding:** Ensured uniform sentence lengths for batch processing. Post padding was used.
- **Label Encoding:** Transformed POS tags into numerical values using LabelEncoder().
- **Sample Weighting:** Addressed padding effects by weighting valid tokens.

2. Model Design

We implemented four recurrent deep learning models:

1. **RNN:** Basic architecture of Recurrent Neural Network, however it is prone to vanishing gradients in long sequences[1].
2. **GRU:** Optimized version of LSTM with fewer parameters. It is faster to train than LSTM [4]. Implements gating mechanisms to reduce vanishing gradients[4].
3. **LSTM:** More powerful than regular RNNs for long-range dependencies[3]. Implements a cell-state structure[3].

4. **BiLSTM:** Both past and future contexts are captured, enhancing classification accuracy[5]. Consists of two LSTMs, one of which is a forward model and the other of which is a backward model[5].

3. Hyperparameter Tuning

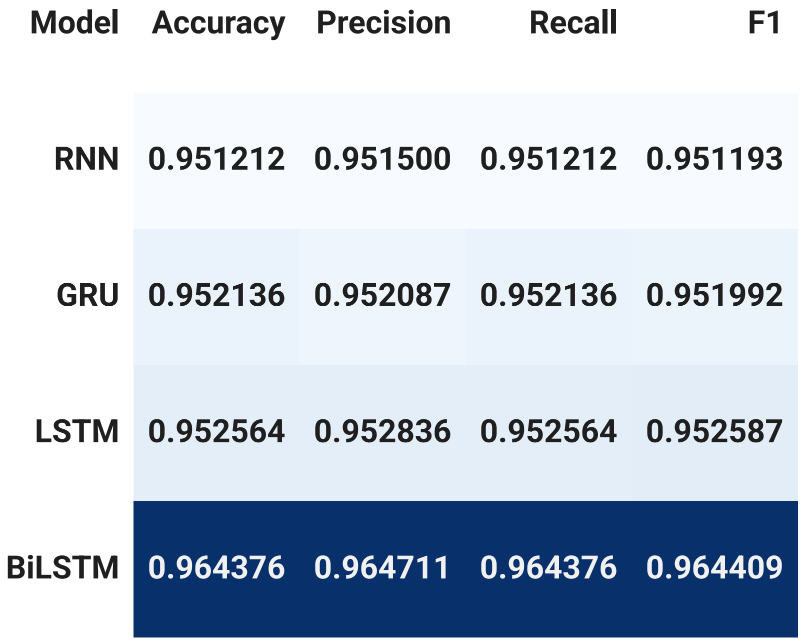
We conducted **grid search tuning** with different configurations for:

- **Units**: *[40, 81, 162]* (81 is the maximum length of a sentence in the training set)
- **Learning Rate:** [*0.01, 0.001*] for RNN [due to vanishing gradient problem], [*0.001, 0.0005*] for the rest
- **Batch Size:** [*64,128*]
- **Epochs:** [*30,50*]

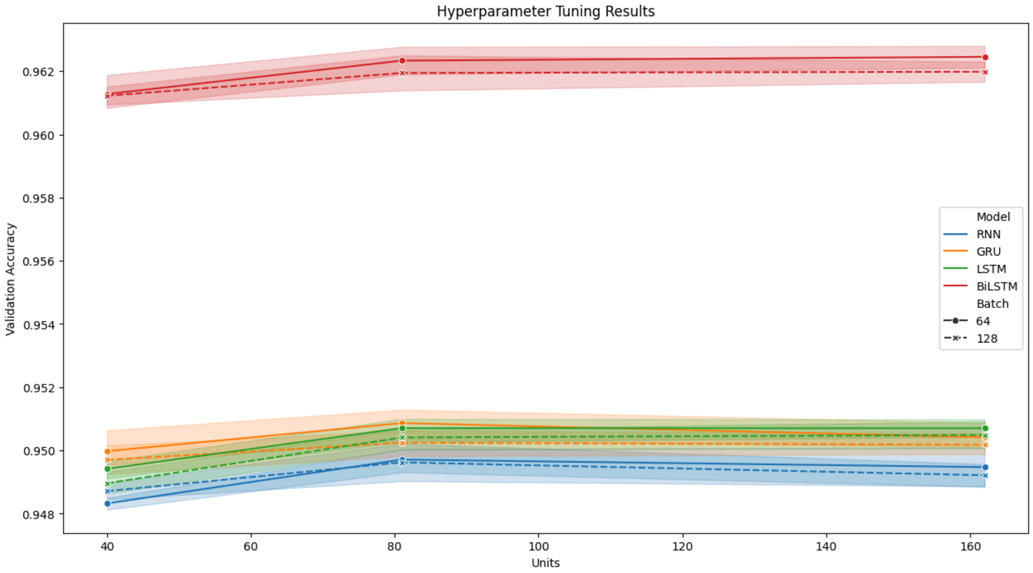
Early stopping ensured optimal training without overfitting. Dropout was not implemented in this project due to the shallowness of the models.

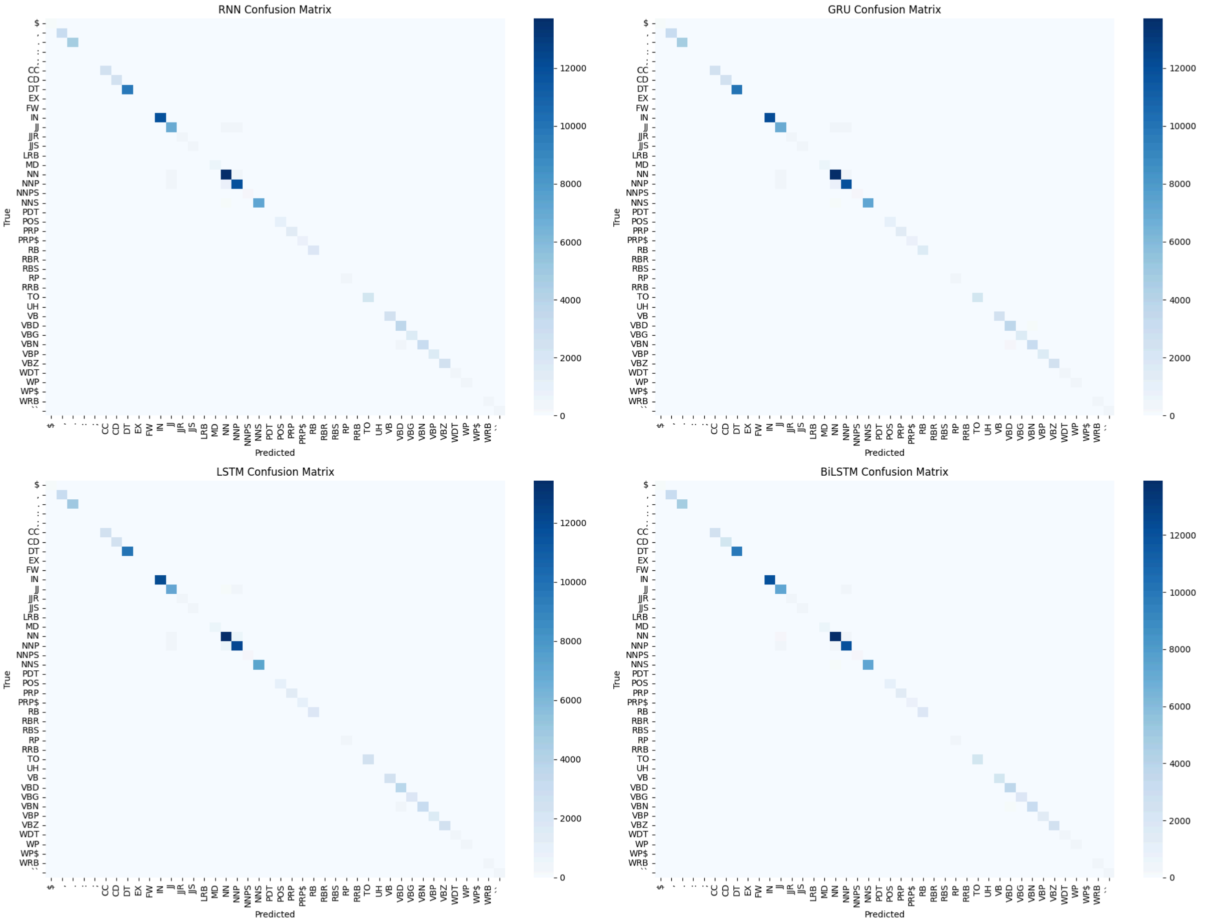
Results:

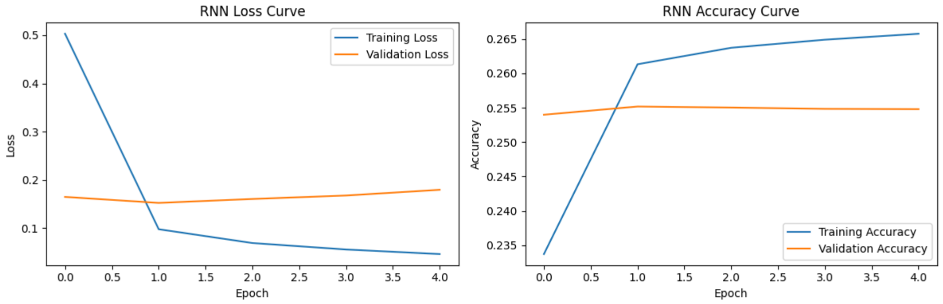
Final Model Comparison:

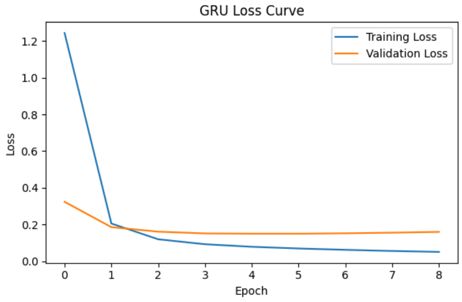


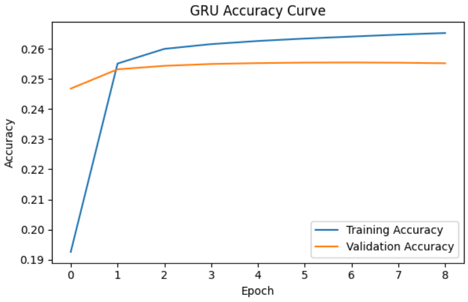
Model	Accuracy	Precision	Recall	F1
RNN	0.951212	0.951500	0.951212	0.951193
GRU	0.952136	0.952087	0.952136	0.951992
LSTM	0.952564	0.952836	0.952564	0.952587
BiLSTM	0.964376	0.964711	0.964376	0.964409

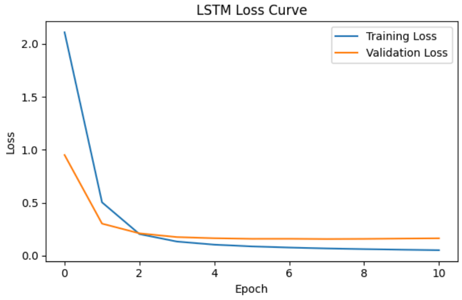

Hyperparameter Tuning Results
Validation Accuracy
0.948
0.950
0.952
0.954
0.956
0.958
0.960
0.962
Units
40
60
80
100
120
140
160
Model
RNN
GRU
LSTM
BiLSTM
Batch
64
128

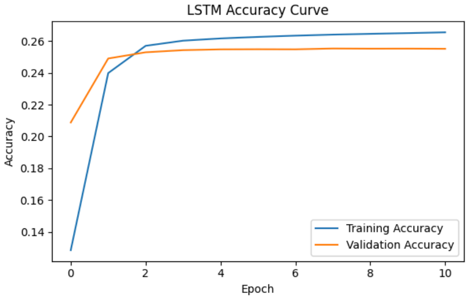

RNN Confusion Matrix
GRU Confusion Matrix
LSTM Confusion Matrix
BiLSTM Confusion Matrix
True
Predicted
$, . : CC CD DT EX FW IN JJ JJR JJS LRB MD NN NNP NNPS NNS PDT POS PRP PRP$ RB RBR RBS RP RRB TO UH VB VBD VBG VBN VBP VBZ WDT WP WP$ WRB
0
2000
4000
6000
8000
10000
12000

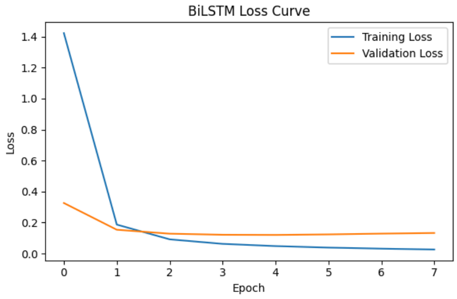

RNN Loss Curve
Loss
Epoch
Training Loss
Validation Loss
0.1
0.2
0.3
0.4
0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
RNN Accuracy Curve
Accuracy
Epoch
Training Accuracy
Validation Accuracy
0.235
0.240
0.245
0.250
0.255
0.260
0.265

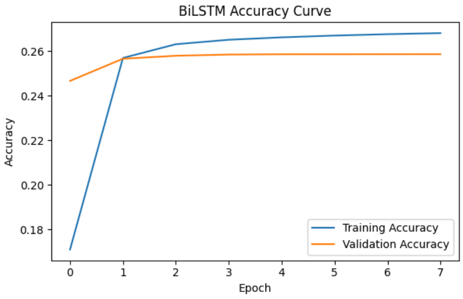












Observations on Model Evaluation Reports:

- BiLSTM achieves the highest accuracy and F1-score among all models, confirming that bidirectional processing improves context understanding. Most common POS tags (NN, DT, IN, JJ, NNS) performed exceptionally well.
- GRU performed slightly better than RNN and LSTM, with the highest accuracy and F1-score.
- RNN performed well, but traditional recurrent networks often struggle with longer dependencies.
- LSTM was comparable to GRU but slightly less efficient, reinforcing the notion that GRUs can be more efficient for certain sequence tasks.
- Rare tags (FW, UH, PDT) had minimal recall, likely due to insufficient training samples.
- Some punctuation classes (:) might show variability.
- All models perform exceptionally well (above 95% accuracy), demonstrating strong robustness in POS classification.
- All models showed remarkably similar confusion matrices, with the most frequent tags being Nouns and Verbs for all models.

Hyperparameter tuning visualization:

- Validation accuracy increases with more units across all models. BiLSTM's highest validation accuracy consistently demonstrates and strengthens its best context awareness.

- GRU and LSTM perform similarly, though GRU may be slightly more efficient.
- Batch size has little effect as most gains are unit count driven.

Loss and Accuracy Curve Analysis:

- LSTM and BiLSTM maintain the lowest validation loss and highest accuracy, indicating strong generalization across the dataset.
- RNN struggles with convergence, showing fluctuating loss patterns. A large gap between training & validation accuracy is signaling overfitting.

Conclusion:

This study successfully demonstrated the effectiveness of deep learning architectures for POS tagging. BiLSTM emerged as the best model, achieving 96.43% accuracy, showcasing superior context-awareness compared to RNN, GRU, and LSTM.

Limitations

In spite of the robust performance of deep learning models in POS tagging, a few challenges were encountered during evaluation. Rare POS tags were affected by data sparsity, resulting in lower recall scores. As there were a few training samples for some classes, the models were not able to generalize their predictions well for these rare tags. Misclassifications were also found with punctuation and special characters, especially with the colon (:) tag, where inconsistency was seen with different architectures. This highlights the need for enhanced token context awareness in punctuation-related labels. Another crucial limitation was computational overhead, specifically with BiLSTM, where increased GPU capacity and memory usage were required compared to simple models like RNN and GRU. This can limit use for large-scale datasets or applications on low-utility hardware. Likely overfitting dangers were also identified—early stopping did cut back on this.

References:

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” Technical Report, California Univ. San Diego, La Jolla, CA, 1985.

[2] M. I. Jordan, “Serial order: A parallel distributed processing approach,” Technical Report, Dept. Computer Science, University of California, San Diego, CA, 1986.

[3] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[4] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1724–1734.

[5] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.