Project Report

Abstract

This project explores the task of Parts Of Speech (POS) tagging using POS Dataset1. We implemented and compared four sequential models: Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional LSTM (BiLSTM). After exploring the dataset and applying preprocessing steps, each model was trained and evaluated using classification report, accuracy, F1-score (both macro and weighted), precision, recall and confusion matrix. Among the models, BiLSTM achieved the highest performance based on accuracy (65%), F1 Macro (48%) and F1 Weightened (65%).

I. Introduction

Part-of-Speech (POS) tagging is a fundamental task in Natural Language Processing (NLP) that involves assigning grammatical categories (such as noun, verb, adjective, etc.) to each word in a sentence. It serves as a critical preprocessing step for a wide range of NLP applications, including syntactic parsing, information extraction, machine translation, and sentiment analysis. (Jurafsky & Martin, 2023). Traditional approaches to POS tagging relied on hand-crafted rules or statistical models like HMMs and CRFs, but these methods often struggled with contextual understanding and long-term dependencies. With the advent of deep learning, RNNs and their variants have become popular due to their ability to model sequential data effectively. This project aims to compare four RNN-based architectures—Simple RNN, LSTM, GRU, and BiLSTM—to assess their strengths and weaknesses in POS tagging tasks. The motivation is to understand how different memory mechanisms and context-handling capabilities influence tagging accuracy and overall performance.

II. Methodology

A. Exploratory Method Analysis

POS Dataset 2 contains sentences with tokens labeled by their corresponding POS tags. The data was split into training and testing sets (80:20). Initial analysis revealed:

a. 19,183 sentences for training
b. 4,796 sentences for testing
c. The most frequent tags include NN, NNP, IN, DT and JJ indicating a typical distribution found

B. Data Preprocessing

The following steps were taken to preprocess the data:

a. Tokenized the sentences using Keras' Tokenizer, with settings to lowercase the text and handle out-of-vocabulary words using a special <OOV> token.
b. Encoded the POS tag sequences using LabelEncoder from sklearn to convert categorical labels into integers.

c. Transformed the integer-encoded POS tags into one-hot vectors, making them suitable for categorical_crossentropy loss and applied padding to both input and output so that all sequences have the same length.
d. Loaded pre-trained Glove embeddings *(glove.6B.100d.txt)* and constructed an embedding matrix to initialize the embedding layer
e. Excluded padding tokens from evaluation metrics by filtering out labels with value 0, so accuracy and F1-score calculations are not skewed.

C. Model Architectures :

1. **Simple RNN:**
Simple Recurrent Neural Networks (RNNs) maintain a hidden state while processing sequences token by token. However, they suffer from the vanishing gradient problem, which limits their ability to capture long-term dependencies in language sequences (Elman, 1990).

Structure:
a. 1 Embedding Layer which converted word indices into 100-dimensional GloVe Vectors.
b. Standard recurrent layer with 64 units.
c. Applied a Dense layer with Softmax activation in each step individually.
d. Converged at epoch 10.

2. **Long Short-Term Memory (LSTM):**
LSTM networks introduce memory cells and gating mechanisms that help them overcome the vanishing gradient issue and capture long-range dependencies (Hochreiter & Schmidhuber, 1997). They are effective at learning when to remember or forget information over time
Structure:
a. 1 Embedding layer (non-trainable, using GloVe)
b. LSTM layer with 64 units
c. 1 TimeDistributed Dense layer with softmax for POS tag prediction
d. Converged at epoch 10

3. **Gated Recurrent Unit (GRU):**
GRUs simplify the LSTM structure by merging the input and forget gates into a single update gate, making them faster to train while maintaining comparable performance (Cho et al., 2014). This makes GRUs particularly suitable for applications with limited computational resources.

Structure:
a. 1 Embedding layer
b. 1 GRU layer with 64 units
c. 1 TimeDistributed Dense layer
d. Converged at epoch 10

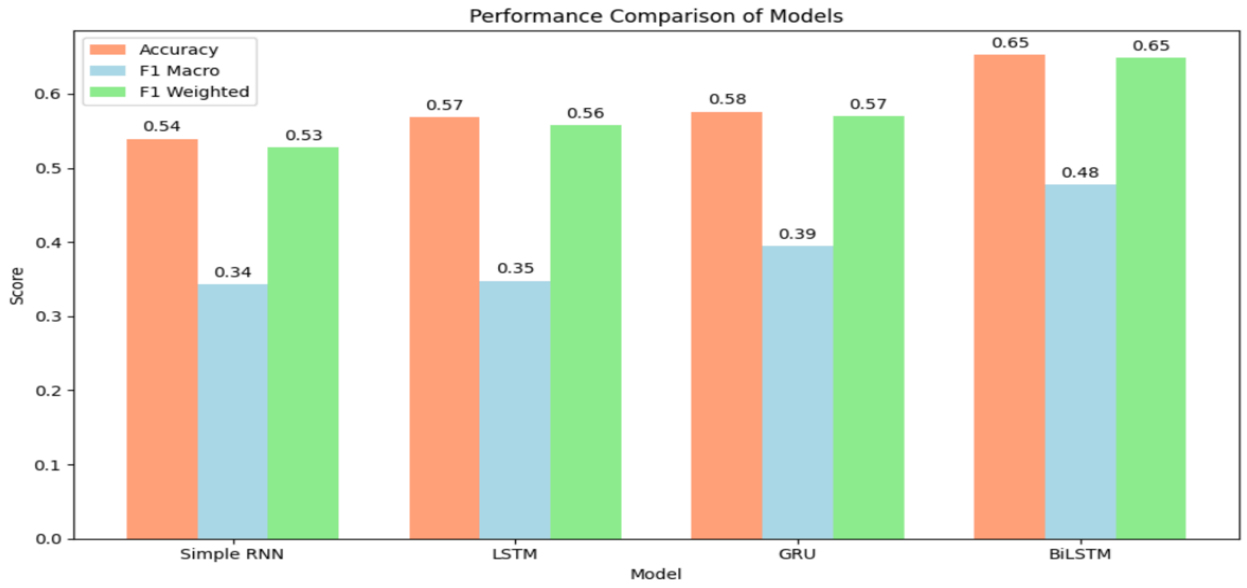
4. **BiLSTM (Bidirectional LSTM):**
BiLSTMs extend the LSTM architecture by adding a second LSTM that processes the input in the reverse direction. This allows the network to consider both past and future context (Schuster & Paliwal, 1997), significantly improving performance on tasks where surrounding context matters.
Structure:
a. 1 Embedding layer
b. 1 Bidirectional LSTM layer with 128 units
c. 1 TimeDistributed Dense layer
d. Converged at epoch 20

Each model was trained using categorical cross-entropy loss and the Adam optimizer, with
tuning over learning rate, batch size and epochs.

III. Results

Summary:

Model	**Accuracy**	**F1 Score Macro**	**F1 Score Weighted**
RNN	0.54	0.34	0.53
LSTM	0.57	0.35	0.56
GRU	0.58	0.39	0.57
BiLSTM	0.65	0.48	0.65



The confusion matrix was used to analyze the prediction accuracy of each model by comparing
the actual POS tags with the predicted ones.

IV. Conclusion

This study demonstrates the effectiveness of sequential models for token-level POS tagging, especially the BiLSTM model, which outperforms others using bidirectional context. Simple RNN, despite being the most basic architecture, delivered competitive results in accuracy but struggled with underrepresented POS classes due to its inability to capture long-range dependencies. LSTM and GRU, while performing better than Simple RNN, did not outperform BiLSTM in this task, suggesting that bidirectional context is particularly valuable in POS tagging.While the results are promising, future work could explore transformer-based architectures like BERT or integrate character-level embeddings to further enhance tag recognition, especially for rare or unknown tokens.

REFERENCES:

1. Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed., draft). Stanford University. https://web.stanford.edu/~jurafsky/slp3/

2. Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179–211 https://doi.org/10.1207/s15516709cog1402_1

3. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

4. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP*. https://arxiv.org/abs/1406.1078

5. Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. https://doi.org/10.1109/78.650093