# CSE440: Lab Project Report

# Project Title:

## Group: 03

## Section : 03

## Semester: Summer 2025

| ID | Name |
|---|---|
| 23241066 | Nafim Rahman |
| 24141071 | Nawroz Haseen Tumul |
| 23241051 | Salim Miah |

# Table of Contents

**Table of Contents** .......... **2**

**Abstract** .......... **3**

**Introduction** .......... **3**

**Methodology** .......... **3**

**Results** .......... **4**

**Conclusion** .......... **5**

**References** .......... **6**

# Abstract

This project focuses on developing a token-level multi-class classifier for Part-of-Speech (PoS) tagging, a fundamental task in Natural Language Processing (NLP). The classifier processes sentences, tokenizes them into words, and assigns a PoS tag to each token. Several deep learning models were explored, including Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Bidirectional LSTM (BiLSTM). The models were evaluated using accuracy and weighted F1-score metrics. The RNN model demonstrated the highest performance, achieving an accuracy of 0.263 and a weighted F1-score of 0.251, followed closely by the BiLSTM model with slightly better accuracy (0.269) and a weighted F1-score of 0.261. In contrast, the LSTM model underperformed with an accuracy of 0.149 and a weighted F1-score of 0.111. The results suggest that while the models did not achieve high performance, the RNN and BiLSTM models were the most effective, with improvements potentially possible through dataset expansion and the use of pretrained embeddings.


# Introduction

Parts of Speech (PoS) tagging is one of the core tasks of Natural Language Processing. The simple task of tagging words with the parts of speech can further shorten the gap between the machines' understanding of humans' ever-so ambiguous use of language. This simple function can be implemented in NLP applications such as sentiment analysis, machine translation, and information retrieval (GeeksforGeeks, 2024).

The objective of this project is to develop a token-level multi-class classifier, which will be able to take in a sentence as input, break it down into tokens (words), and assign a PoS tag to each of those tokens.

# Methodology

**Exploratory Data Analysis (EDA)**: To begin with, we did a basic overview of the data to view the first few sentences and the basic structure of the columns and rows, as well as the number of rows and columns, their names, and their data types. Missing value analysis was done to find any missing or null values. Then we run several analyses on both the sentences (and their tokens) and the PoS tags. We analyzed the lengths of sentences, vocabulary size, distribution and frequency of the tokens and the tags and presented the top ones, which was done to check for the balance of the dataset. Furthermore, we ran an outlier detection to locate larger or smaller sentences than the average range of the sentence size and a co-occurrence analysis of tokens for creating embeddings of tokens. Finally, we made word cloud representations of both the tokens and the PoS tags.

**Preprocessing**: For preprocessing, we created a preprocessing function that consisted of the following:

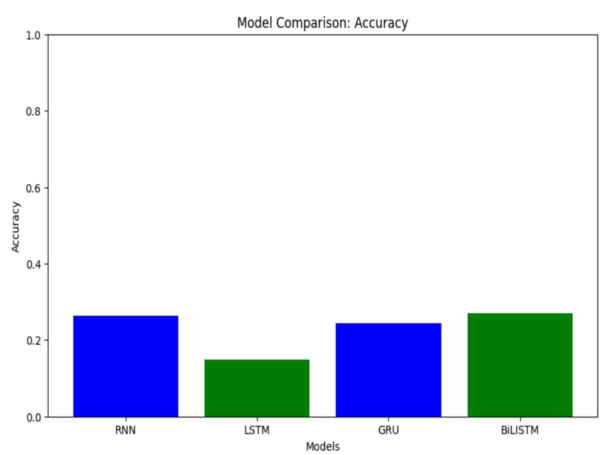
- Tokenizer: Tokenized the sentences into tokens (words).
- Padding: Added padding to both sequences of tokens and corresponding embedded PoS tags to ensure uniform input for the models by adding 0 values.
- Encoding the PoS tags: Assign a numeric value to each tag.

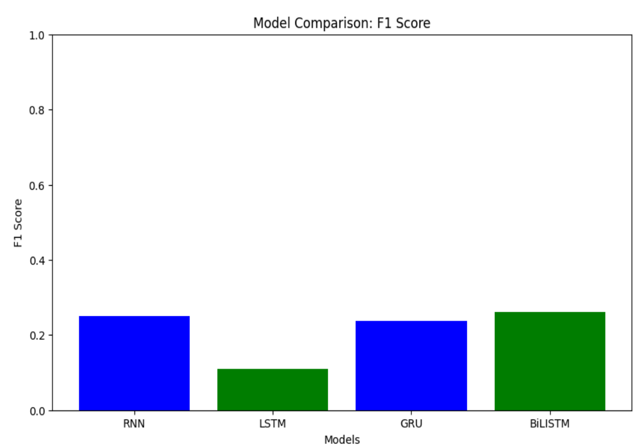
**Model Architecture:**

- RNN: Uses a simple recurrent layer to process sequences step-by-step, followed by a dense softmax layer for POS tag prediction.
- LSTM: Replaces SimpleRNN with LSTM layers for better capture of long-term dependencies via gating mechanisms, then predicts tags with softmax.
- GRU: Uses GRU layers (simplified gating) for efficient sequence processing, ending with a softmax classification layer.
- BiLSTM: Employs bidirectional LSTM to analyze sequences both forward and backward, enhancing context awareness before softmax prediction.

All models shared the same embedding input and output structure.

# Results



The RNN Model achieves the highest accuracy (0.263) and F1-score (weighted: 0.251), outperforming the other models in both metrics. The BiLSTM Model comes in second, with slightly better accuracy (0.269) and a higher weighted F1-score (0.261) than the GRU and LSTM models. The GRU Model shows moderate performance, with an accuracy of 0.244 and a weighted F1-score of 0.238, which is just slightly below the RNN Model. The LSTM Model exhibits the lowest performance across both metrics, with an accuracy of 0.149 and a weighted F1-score of 0.111, indicating that it struggles compared to the other models in this particular task. Overall, while none of the models achieve particularly high performance, the RNN and BiLSTM models are comparatively more effective based on the evaluation metrics.

# Conclusion

In conclusion, RNN & BiLSTM > GRU > LSTM. One of the major factors affecting accuracy was the size of the dataset and the fact that we generated embeddings based on a relatively smaller amount of data. A bigger dataset would produce more accurate results and would provide the models more exposure to handle unknown words. Relatively smaller dataset and lack of pretrained embeddings can cause rare tags to be misclassified. Models like LSTM and BiLSTM are slower compared to RNN and GRU. Using pre-trained word vectors such as GloVe by standard can improve model accuracy. Also, using more powerful models such as the transformer model can be promising. Training for a specific context and using the model for the aforementioned contexts can also be an approach.

# References

GeeksforGeeks. (2024, January 3). *POS(Parts-Of-Speech) Tagging in NLP*. GeeksforGeeks. https://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging/