**Exercise for Unit 7**

Name: CRISTOPHER IAN ARTACHO Date: May 18, 2025

Name: DHOMINICK JOHN BILLENA

Year and Section: BSCS 3A

*Note: Upload your sources in a public GitHub repository and put the link in the document.*

1. (70 points) Choose two NLP tasks:
   1. PoS Tagging
   2. NER Recognition
   3. Text Prediction
   4. Sentiment Classification

Create 2 models, 1) the traditional (machine learning technique appropriate for the task) used, and 2) train a neural network (Feedforward, RNN, LSTM, also appropriate for the task).

For the corpus, you can use any appropriate dataset for the NLP Task.

You can use Python libraries suited for neural networks.

1. (30 points) Evaluate each model and compare its performance. Which of the models has performed the best?

* PoS Tagging: I utilized a dataset from CommonLit which is named as the CLEAR dataset which categorized the sentences based on the grammatical rules it has and the number of noun-verb sequence. It is perfect for PoS Tagging and Analysis. The results show that while the HMM model trained much faster (1.53s vs. 622.54s for BiLSTM), the BiLSTM achieved higher accuracy (96.19% vs. 94.97% for HMM). The BiLSTM also performed inference more efficiently, taking only 6.38s compared to the HMM's 46.74s. This demonstrates the trade-off between training time and performance - the BiLSTM requires significantly more training resources but delivers superior accuracy and faster inference, making it the better choice for applications where prediction speed and accuracy outweigh initial training costs.
* Sentiment Analysis: I utilized a dataset from Kaggle consisting of sentences that can be categorized into three distinct categories: Positive, Neutral and Negative. I trained Multinomial Naïve Bayes and LSTM models. The Multinomial Naïve Bayes utilized tfidf in order to count the number of occurrences of the words. It managed to score an accuracy of .72 with an F1 score of .7228 when evaluating. The LSTM model on the other hand, was trained using the parameters: *loss*='categorical\_crossentropy', *optimizer*='adam', *metrics*=['accuracy']. It managed to only get an accuracy of .36 and F1 score f .53. With the dataset provided, Naïve Bayes was better than LSTM.

1. Put the GitHub repo link here:

* https://github.com/Cristopher-Artacho-WVSU/NLP\_Assignment7