DSC4213 - Neural Networks and Deep Learning

Sequence Model/Recurrent Neural Network Models for Personal Health Mention Classification

Pasindu Kurukulasuriya

S/19/421

Department of Statistics and Computer Science
University of Peradeniya

1. Introduction

The creation and assessment of two recurrent neural network (RNN) models—Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (Bi-LSTM)—for categorizing tweets as "personal health mentions" or "non-personal health mentions" is described in this report. Analyzing and contrasting how well these models perform on the provided dataset is the goal.

2. Methodology

2.1 Data Loading and Splitting

The data is loaded from the *phm_train.csv* and *phm_test.csv* files. The 'tweet' column serves as the input features (x_data), and the 'label' column serves as the target variable (*y_data*). The training data is further split into training and validation sets (80% train, 20% validation) to monitor model performance during training.

The *load_dataset* function preprocesses the tweets by converting them to lowercase.

2.2 Tokenization and Padding

- **Tokenization**: The *Tokenizer* from *tensorflow.keras.preprocessing.text* is used to convert text tweets into sequences of integers, where each integer represents a unique word. The tokenizer is fitted only on the training data to avoid data leakage.
- **Padding**: Tweets vary in length. To feed them into a neural network, they need to have a uniform length. *pad_sequences* is used to achieve this. The *maximum length* is determined by the mean length of tweets in the training set (approximately 16 words). Shorter sequences are padded with zeros, and longer sequences are truncated.

- The total number of unique words after tokenization is 11344.

3. Model Architecture

Two deep learning models are implemented for this classification task: LSTM and Bidirectional LSTM. Both models utilize an embedding layer to represent words as dense vectors, followed by recurrent layers and a final dense output layer with a sigmoid activation for binary classification.

3.1 LSTM Model

The LSTM model consists of :

- **Embedding Layer:** Maps each word in the vocabulary to a dense vector of *EMBED_DIM* (32) dimensions. The *input_length* is set to *max_length*.
- **LSTM Layer:** A standard LSTM layer with *LSTM_OUT* (64) units. This layer processes the sequence of embedded words, capturing long-term dependencies.
- **Dense Output Layer:** A single neuron with a sigmoid activation function, outputting a probability score for the "personal health mention" class.

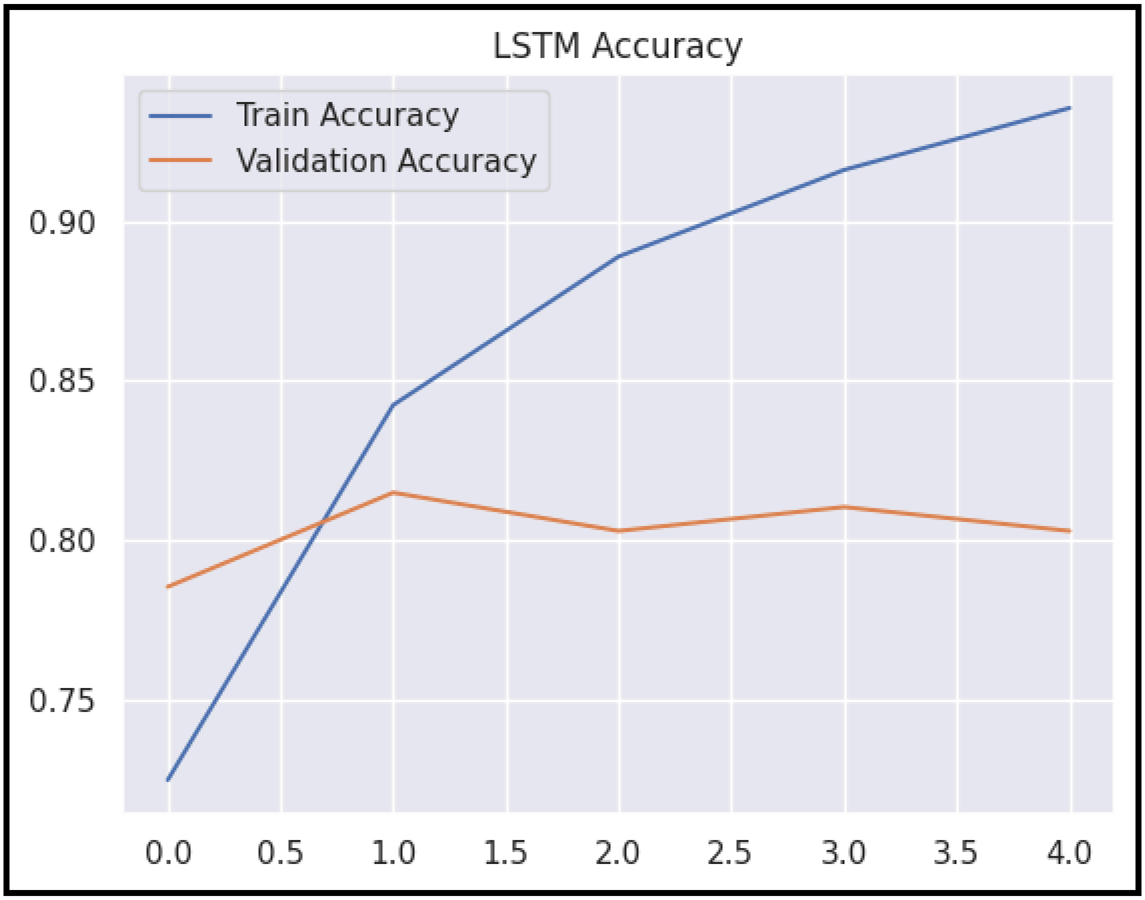
Model Compilation: The model is compiled with the *Adam* optimizer, *binary_crossentropy* loss function (suitable for binary classification), and *accuracy* as the evaluation metric.

Training: The model is trained for 5 epochs with a batch size of 128. *ModelCheckpoint* is used to save the best model based on validation accuracy, and *EarlyStopping* is implemented to stop training if the validation loss does not improve for 3 consecutive epochs.

Testing:

```
105/105 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step
Correct Prediction: 2711
Wrong Prediction: 620
Accuracy: 81.38697087961573
```

Results: The following plot visualizes the training and validation accuracy and loss over the epochs for the LSTM model.



3.2 Bidirectional LSTM Model

The architecture is similar to the LSTM model but replaces the *LSTM* layer with a *Bidirectional(LSTM)* layer:

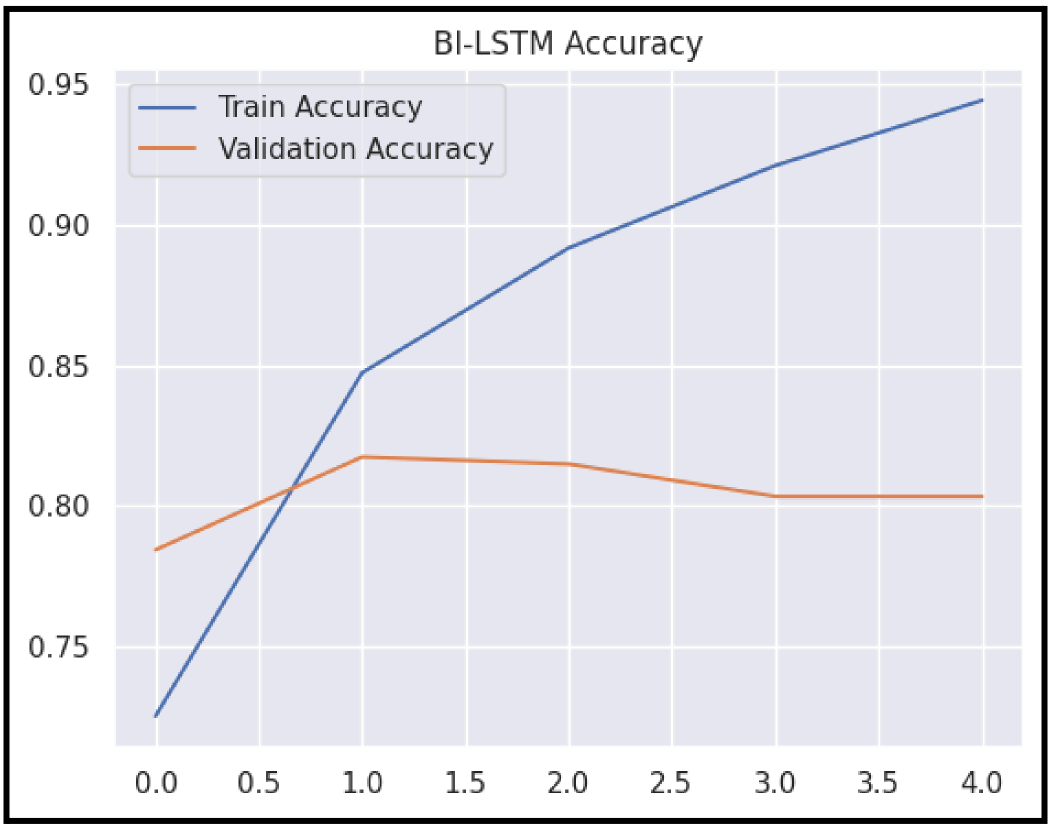
- **Embedding Layer:** Same as the LSTM model.
- **Bidirectional LSTM Layer:** A *Bidirectional* wrapper around an *LSTM* layer with *LSTM_OUT* (64) units.
- **Dense Output Layer:** Same as the LSTM model.

Model Compilation and Training: The Bi-LSTM model is compiled and trained using the same parameters as the LSTM model for fair comparison.

Testing:

```
105/105 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step
Correct Prediction: 2707
Wrong Prediction: 624
Accuracy: 81.26688682077454
```

Results: The following plot visualize the training and validation accuracy and loss over the epochs for BI-LSTM model.

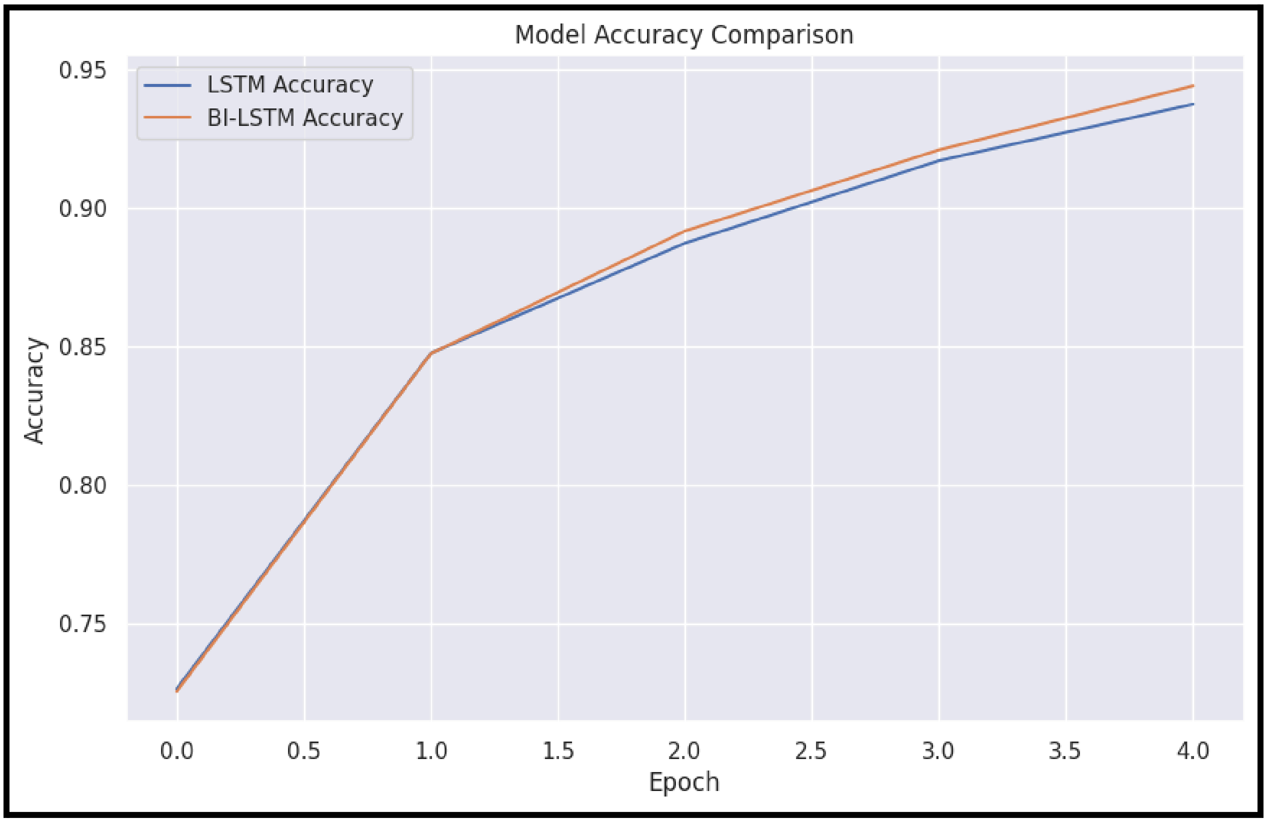


4. Performance Comparison

To compare the performance of the LSTM and Bi-LSTM models, we will analyze their accuracy and loss metrics on both the training and validation datasets across epochs.

4.1 Accuracy and Loss Plots

The following plots visualize the Model accuracy over the epochs for both models.

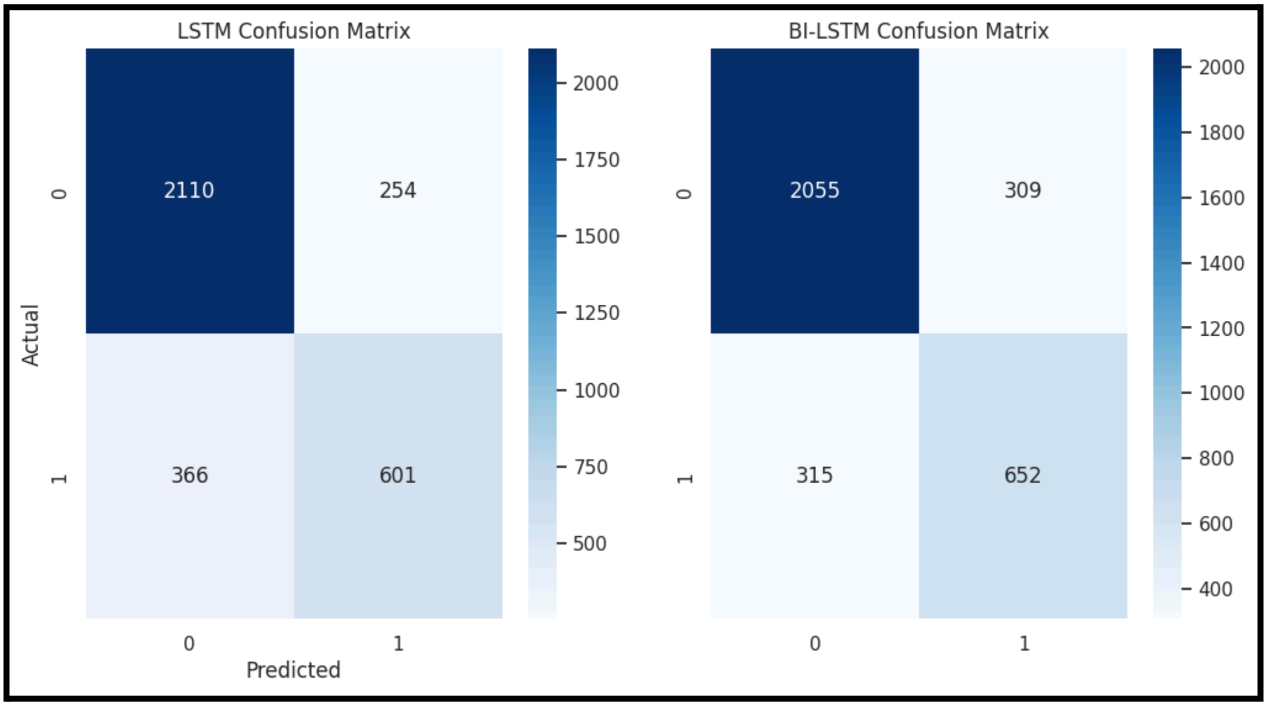


4.2 Performance Summary Table

Model	Test Accuracy
LSTM	81.38697087961573

Bi-LSTM	81.26688682077454

4.3 Confusion Matrix



4.4 Overall Findings

When comparing LSTM and Bi-LSTM models for classifying health-related tweets, the simpler LSTM model actually performed better in this project. While Bi-LSTM can read text both forwards and backwards for full context, for our specific data, LSTM achieved higher accuracy and lower error rates, making it more reliable for new tweets. This shows that sometimes, a simpler model can be more effective, highlighting that choosing the right model is crucial for getting the best results in understanding language.

5. Conclusion

Both LSTM and Bi-LSTM models are effective in classifying personal health mentions in tweets. However, for this specific task and dataset, the LSTM model proved to be more effective, leading to better overall performance. Further improvements can still be achieved through more extensive hyperparameter tuning and exploring advanced techniques like pre-trained word embeddings or more complex architectures, but the LSTM stands out as the preferred choice here.