News Category Classification Using LSTM: Report

1. Introduction

This report details the process of classifying news articles into categories using a Long Short-Term Memory (LSTM) model. The dataset consists of news article titles, and the objective is to classify these titles into one of four categories: Business (b), Technology (t), Entertainment (e), and Health (m).

2. Data Preprocessing

2.1. Dataset Overview

The original dataset contained 422,419 rows and multiple columns. For this task, only the TITLE and CATEGORY columns were used:

- **Titles:** Contain the headlines of news articles.
- **Categories:** Represent the labels/classes for the news articles, identified as b, t, e, and m.

2.2. Data Balancing

To ensure balanced training, 45,000 samples were randomly selected from each category, leading to a total of 180,000 samples:

- **Classes:** Business (b), Technology (t), Entertainment (e), Health (m)
- **Class Distribution:** 45,000 samples per class

2.3. Label Encoding

The categorical labels (b, t, e, m) were encoded as integers (0, 1, 2, 3) and then converted into one-hot encoded vectors, suitable for multi-class classification:

- **Business (b) → 1**
- **Technology (t) → 2**
- **Entertainment (e) → 0**
- **Health (m) → 3**

2.4. Tokenization and Padding

The TITLE column was tokenized using Keras' Tokenizer class, retaining the top 8,000 most frequent words. The sequences were padded to a maximum length of 130 tokens to ensure uniform input size.

Summary of Preprocessing:

- **Unique Tokens Found:** 52,589
- **Tokenized and Padded Sequences:** Shape (180,000, 130)
- **Train-Test Split:** 75% training, 25% testing

3. Model Architecture

An LSTM model was designed to classify the news article titles. The architecture is as follows:

- **Embedding Layer:** Converts the tokenized sequences into dense vectors of fixed size (128 dimensions).
- **Spatial Dropout:** Applied to prevent overfitting, with a dropout rate of 70%.
- **LSTM Layer:** A single LSTM layer with 64 units and 70% dropout and recurrent dropout rates.
- **Dense Output Layer:** A dense layer with 4 output neurons and a softmax activation function for multi-class classification.

Model Summary:

- **Total Parameters:** 1,073,668
- **Trainable Parameters:** 1,073,668
- **Non-Trainable Parameters:** 0

4. Training and Evaluation

4.1. Training

The model was trained for 10 epochs with a batch size of 128, using the Adam optimizer and categorical cross-entropy loss. Early stopping was employed to monitor the validation loss and prevent overfitting:

- **Training Accuracy:** Started at 81.11% and improved to 92.03% over 5 epochs.
- **Validation Accuracy:** Improved from 90.90% to 92.64%.

4.2. Testing

After training, the model was evaluated on the test set:

- **Test Set Loss:** 0.2166
- **Test Set Accuracy:** 92.64%

Performance Metrics:

- The model showed strong performance with high accuracy on both training and validation sets.
- The early stopping mechanism ensured that the model did not overfit to the training data.

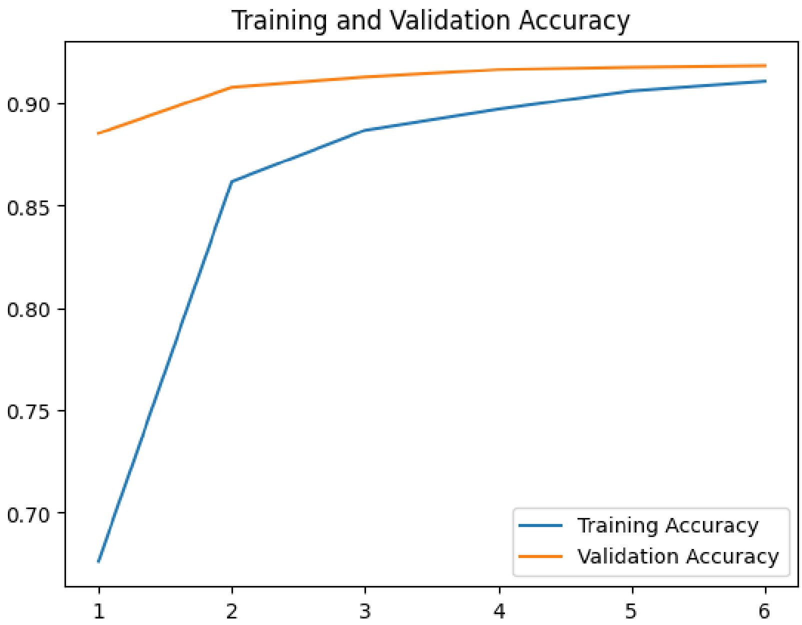
5. Results Visualization

Two key plots were generated to visualize the training process:

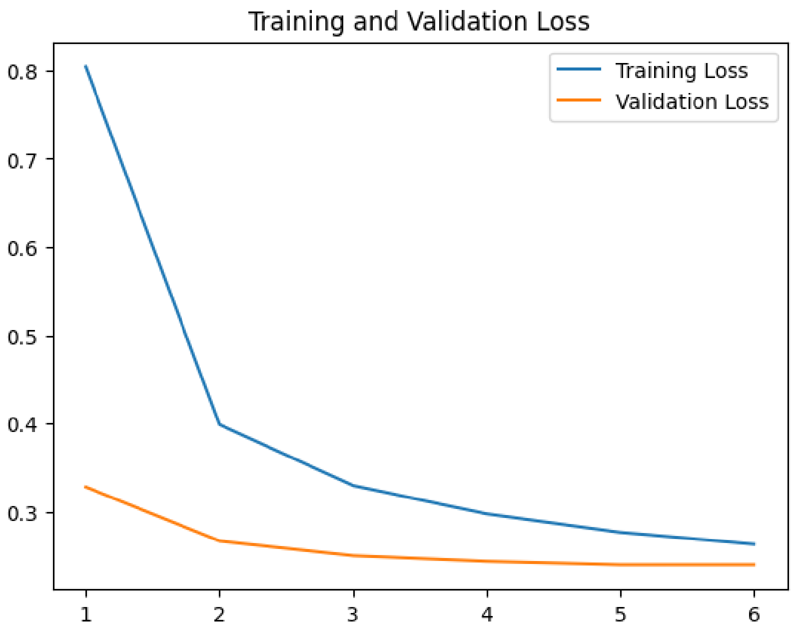
- **Training and Validation Accuracy:**
    - The plot shows a consistent increase in accuracy over the epochs, with the validation accuracy closely following the training accuracy, indicating a well-generalized model.
- **Training and Validation Loss:**
    - The loss decreased steadily over the epochs, with the validation loss also showing a similar trend, further indicating the model's robustness.

Visualizations:

1. **Training and Validation Accuracy Plot**



2. **Training and Validation Loss Plot**



6. Conclusion

The LSTM model successfully classified news article titles into four categories with a high degree of accuracy. The combination of data preprocessing, balanced sampling, and a well-tuned LSTM architecture resulted in a model that generalized well to unseen data. The model can be further improved by experimenting with deeper architectures, different tokenization strategies, or even pre-trained word embeddings.

This classification approach demonstrates the effectiveness of LSTM networks in handling sequence data, particularly for text classification tasks where context and order are crucial.