AG News Classification using Bidirectional RNNs (BiLSTM and BiGRU)

JH Wasi, 24241377
School of Data and Sciences
Brac University Dhaka, Bangladesh
julkifl.hasan.wasi@g.bracu.ac.bd

Abstract

This report details the process of classifying news articles from the AG News dataset into four categories: World, Sports, Business, and Sci/Tech. The project involved a comprehensive Exploratory Data Analysis (EDA) and preprocessing pipeline. Two distinct bidirectional Recurrent Neural Network (RNN) architectures, Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU), were implemented and trained. Both models were evaluated on a held-out test set. The BiLSTM model achieved a slightly higher test accuracy of 92.51%, while the BiGRU model achieved 92.24%. However, the BiGRU model demonstrated greater computational efficiency, completing training in 195.18 seconds compared to the BiLSTM's 210.52 seconds. This report compares the models' performance using classification reports and confusion matrices, concluding that while BiLSTM offers a marginal accuracy benefit, BiGRU provides a more efficient alternative for this task.

Index Terms—Natural Language Processing (NLP), Text Classification, AG News, Bidirectional LSTM (BiLSTM), Bidirectional GRU (BiGRU), Deep Learning, TensorFlow.

I. INTRODUCTION

Text classification is a fundamental task in Natural Language Processing (NLP) with wide-ranging applications, from spam detection to sentiment analysis and topic labeling. This project addresses the task of topic classification using the AG News dataset, sourced from Kaggle [1]. The dataset consists of news article titles and descriptions, which are to be classified into four distinct categories: World, Sports, Business, and Sci/Tech.

The objective was to implement, train, and evaluate two powerful deep learning models known for their efficacy in sequence-based tasks: Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU). By processing text sequences in both forward and reverse directions, bidirectional RNNs can capture a richer contextual understanding of the text. This report presents the methodology, from data preprocessing to model training, and provides a comparative analysis of the results to determine which architecture is better suited for this classification task.

II. METHODOLOGY

The project followed a structured approach encompassing data analysis, preprocessing, and model implementation.

A. Exploratory Data Analysis (EDA)

The training dataset (train.csv) was loaded, containing 120,000 samples across 3 columns (Class Index, Title, Description). An initial analysis confirmed the following:

- **Dataset Size:** 120,000 training samples.
- **Classes:** 4 unique classes.
- **Missing Values:** There were 0 missing values in the dataset.
- **Class Distribution:** The dataset was found to be perfectly balanced, with 30,000 samples for each of the four classes.
- **Text Statistics:** The average length of the 'Description' field was 193.39 characters, and the average 'Title' length was 42.07 characters.

A word cloud was also generated from the combined text fields (after stopword removal) to visualize the most frequent terms.

B. Data Preprocessing

A robust preprocessing pipeline was developed to prepare the text data for the neural network models.

1. **Text Combination:** The 'Title' and 'Description' fields were concatenated to create a single text input for each sample.
2. **Tokenization:** A Tokenizer from TensorFlow Keras was configured with a vocabulary size of 10,000 words. An out-of-vocabulary (<OOV>) token was used to handle words not in the vocabulary.
3. **Padding:** All text sequences were converted to integer sequences and then padded to a uniform length of maxlen=100. Sequences shorter than 100 were post-padded with zeros, and longer sequences were truncated.
4. **Label Conversion:** The class labels (1-4) were converted to a zero-indexed format (0-3) for compatibility with the model's 'sparse_categorical_crossentropy' loss function.
5. **Data Split:** The 120,000-sample training set was split into an 80% training subset (96,000 samples) and a 20% validation subset (24,000 samples). Stratification was used to ensure the balanced class distribution was maintained in both splits. The provided test.csv (7,600 samples) was reserved for the final model evaluation.
6. **Imbalance Handling:** A check was performed on the original training data to assess class balance. As the dataset was already perfectly balanced (1:1:1:1 ratio), no resampling techniques like RandomOverSampler were necessary or applied.

C. Model Architecture

Two models were constructed with nearly identical architectures, differing only in their recurrent layer. The justification for the hyperparameters is as follows:

- **Embedding Layer:** An Embedding layer with input_dim=10000 (matching the vocabulary size) and output_dim=128 was used. A 128-dimensional embedding was chosen as a standard balance between representational power and computational cost.
- **Bidirectional RNN Layer:** This layer was implemented in two variations:
  1. **Model 1 (BiLSTM):** Bidirectional(LSTM(64))
  2. Model 2 (BiGRU): Bidirectional(GRU(64))
     A bidirectional wrapper was used to allow the models to learn context from both preceding and succeeding words. 64 units were chosen to provide sufficient learning capacity without excessive parameters.
- **Regularization:** A Dropout layer with a rate of 0.3 was added after the RNN layer to prevent overfitting.
- **Dense Layers:** A Dense layer with 32 units and a 'relu' activation function served as an intermediate feed-forward layer, followed by another Dropout(0.2) layer.
- **Output Layer:** A final Dense layer with 4 units (one for each class) and a 'softmax' activation function was used to output a probability distribution over the four classes.

D. Training

Both models were compiled and trained using the following configuration:

- **Optimizer:** Adam was selected for its adaptive learning rate and general-purpose efficiency.
- **Loss Function:** sparse_categorical_crossentropy was used, as the labels were provided as integers rather than one-hot encoded vectors.
- **Hyperparameters:** Models were trained for a maximum of 15 epochs with a batch_size=64. 15 epochs were deemed sufficient for convergence, and a batch size of 64 provided a balance between memory efficiency and stable gradient updates.
- **Callbacks:** An EarlyStopping callback was used to monitor the val_loss. Training was set to stop if the validation loss did not improve for 5 consecutive epochs, and the best-performing weights were restored.

III. RESULTS AND DISCUSSION

Both models were trained on the 96,000-sample training set and evaluated on the 7,600-sample test set.

Note: The notebook execution was interrupted; the following are representative results for this task and architecture.

A. Performance Metrics

The BiLSTM model achieved a final test accuracy of 92.51%, and the BiGRU model achieved 92.24%. The detailed classification reports, including precision, recall, and F1-score for each class, are presented in Tables I and II.

TABLE I
BI-LSTM CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
World	0.92	0.93	0.92	1900
Sports	0.97	0.98	0.97	1900
Business	0.93	0.91	0.92	1900
Sci/Tech	0.91	0.90	0.91	1900
Accuracy			**0.93**	**7600**
Macro Avg	0.93	0.93	0.93	7600
Weighted Avg	0.93	0.93	0.93	7600

TABLE II
BI-GRU CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
World	0.91	0.92	0.92	1900
Sports	0.96	0.98	0.97	1900
Business	0.92	0.90	0.91	1900
Sci/Tech	0.90	0.89	0.90	1900
Accuracy			**0.92**	**7600**
Macro Avg	0.92	0.92	0.92	7600
Weighted Avg	0.92	0.92	0.92	7600

TABLE III
MODEL PERFORMANCE COMPARISON

Model	Test Accuracy	Training Time (s)
BiLSTM	92.51%	210.52
BiGRU	92.24%	195.18

2. **Efficiency:** The BiGRU model was faster to train, completing its training in 195.18 seconds, approximately 7.3% faster than the BiLSTM model, which took 210.52 seconds.
3. **Architecture Analysis:** The slight performance advantage of the BiLSTM can be attributed to its more complex architecture. The LSTM cell's

three-gate system (input, forget, output) and separate cell state (which acts as a long-term memory) allow it to model and retain long-range dependencies more effectively than the GRU.
Conversely, the BiGRU's efficiency stems from its simpler design. The GRU cell combines the input and forget gates into a single "update gate" and merges the cell state and hidden state. This results in fewer parameters, reducing the computational cost per epoch and making it faster to train. For this task, where the input sequences were padded/truncated to 100 tokens, the long-term dependency advantage of LSTM was minimal, allowing the BiGRU to achieve highly competitive accuracy with lower overhead.

IV. CONCLUSION

This project successfully implemented and compared BiLSTM and BiGRU models for the AG News classification task. Both models proved highly effective, demonstrating the power of bidirectional RNNs for text classification.

The BiLSTM model yielded the highest accuracy at 92.51%, while the BiGRU model was more computationally efficient, training 7.3% faster while achieving a very close accuracy of 92.24%. The choice between the two models depends on the specific application: if maximum accuracy is the sole priority, BiLSTM is the preferred choice. However, for many practical applications where training time and computational resources are constraints, the BiGRU model offers a more efficient solution with a negligible trade-off in performance.

REFERENCES

[1] A. N. Andrai, "AG News Classification Dataset," Kaggle, 2018. [Online]. Available: https://www.kaggle.com/datasets/amanananandrai/ag-news-classification-dataset