HW2 experiment Report

Name:李崇楷 Student ID: 313834006

Code: https://github.com/kailee0422/RNN-Transformer/tree/main/HW2

Introduction

This project focuses on building two models—LSTM and GRU—to classify whether a tweet is related to a real disaster. Using the dataset from a Kaggle competition, I train both models, compare their performance in terms of accuracy and efficiency, and generate predictions for unseen test data. This assignment helps us understand the differences between LSTM and GRU in practical NLP tasks..

Method

This study utilizes the BERT language model in conjunction with LSTM and GRU to perform tweet classification. The overall pipeline is divided into four main stages: data preprocessing, model architecture, training setup, and evaluation. Each step is described in detail below.

Data Preprocessing

The raw data consists of tweet texts, which were cleaned and normalized through several preprocessing steps to enhance the quality of model input:

- Punctuation Removal: All punctuation marks were removed using Python's "string.punctuation".
- URL Replacement: All URLs were replaced with the token "URL".
- HTML Tag Removal: HTML tags such as <div> were removed using regular expressions.
- Non-ASCII Character Removal: All non-printable characters were filtered out.
- Abbreviation Expansion: A custom dictionary was created to expand common online abbreviations. For example: "omg" → "oh my god", "idk" → "i do not know"
- User Mention Replacement: Mentions such as @username were replaced with "USER".
- Number Normalization: All numbers were replaced with the token "NUMBER".
- Emoji and Emoticon Handling: Common emojis and emoticons such as :), :(, <3, and 😭 were replaced with standardized tokens like "SMILE", "SADFACE", "HEART", or "EMOJI".

These preprocessing steps ensured that the text data was clean, uniform, and well-suited for downstream processing by language models.

Model Architecture

To leverage both the semantic understanding of pre-trained language models and the contextual modeling capabilities of sequence models, I propose two model architectures: BERT+LSTM and BERT+GRU. The structure is as follows:

- BERT Encoder: I used the bert-base-uncased model to encode each tweet into a sequence of vectors (each token has 768 dimensions).
- LSTM/GRU Layer: A single-layer LSTM or GRU with a hidden size of 256 was used to process the output from BERT.
- Dropout: A dropout layer with a dropout rate of 0.2 was applied to reduce overfitting.
- Fully Connected Layer + Sigmoid: The final hidden state from the RNN was passed through a fully connected layer followed by a Sigmoid activation to produce a binary classification probability.

Importantly, the BERT parameters were frozen to reduce training complexity and only the RNN layers and classifier head were trained.

Training Setup

The training process used the following configuration:

- Loss Function: Binary Cross-Entropy Loss (BCELoss).
- Optimizer: Adam optimizer with a learning rate of 0.0001.
- Learning Rate Scheduler: ReduceLROnPlateau was employed to reduce the learning rate dynamically when the loss plateaued.
- Batch Size: 32.
- Number of Epochs: 30.
- Ratio of validation: 20%

During training, I recorded the training and testing accuracy, loss per epoch, training time, and GPU memory usage to facilitate a detailed comparison between the LSTM and GRU models.

Evaluation

After training, the models were evaluated on a held-out test set using the following metrics:

- Accuracy
- Precision
- Recall

Additionally, I visualized the training and testing loss and accuracy trends over epochs. These plots provide a comparative analysis of the performance and generalization capabilities of the LSTM and GRU models. I also examined training speed and GPU memory efficiency to assess resource utilization.

Result

	Accuracy	Precision	Recall
LSTM	81.68%	80.53%	**75.19%**
GRU	**83.45%**	**85.01%**	74.27%

Table 1. Val Accuracy, Precision and Recall for Different Model

The above table shows that GRU outperforms LSTM in accuracy and precision, suggesting that GRU is better at correctly identifying tweets related to real disasters with fewer false positives. Although LSTM achieves slightly higher recall, GRU provides a more balanced trade-off between precision and recall. This may be because GRU, with its simpler structure and fewer parameters, is more efficient at capturing short-term dependencies in the tweet texts, leading to faster convergence and better generalization on the validation data.

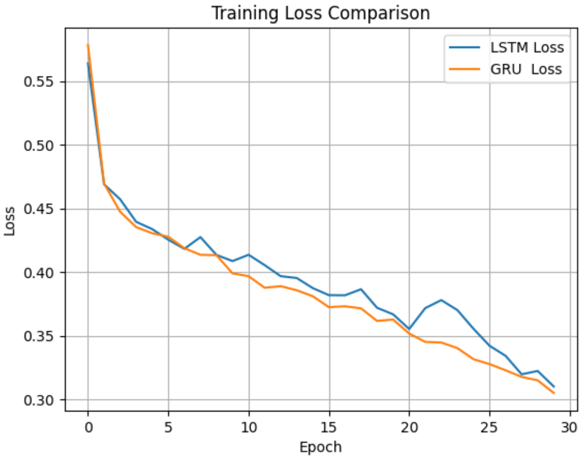

Figure 1. Training loss over epochs for using an LSTM and GPU model.

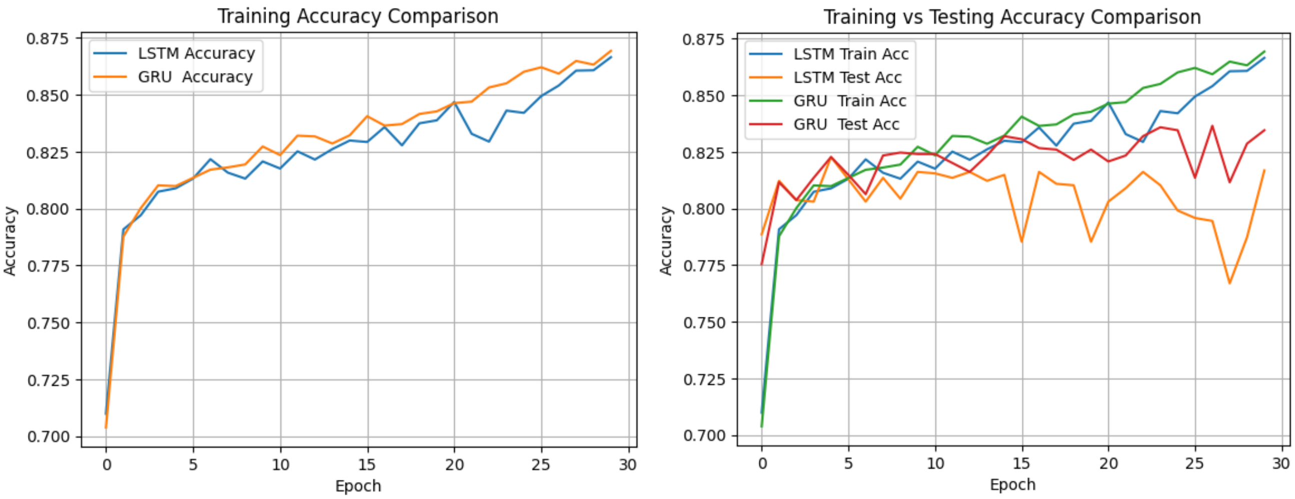

Figure 2 . Training accuracy(left) and validation accuracy compared with train accuracy(right) over epochs for using an LSTM and GPU model.

	Training Speed	GPU Memory Usage	Num of Parameter
LSTM	26.29 s	**1435.24 MB**	110,533,121
GRU	**26.12 s**	1445.26 MB	**110,270,465**

Table 2. Training Speed and memory compared with an LSTM and GRU model.

As shown in the table, GRU has fewer parameters than LSTM, which aligns with its simpler architecture. Despite this, GRU exhibits slightly higher GPU memory usage, possibly due to differences in implementation or memory allocation during training. In terms of training speed, GRU is marginally faster, suggesting that the model's simpler structure allows for more efficient computation despite the slightly higher memory usage. Overall, GRU offers a good trade-off between model complexity and performance.

	Accuracy	Precision	Recall	F1 Score
LSTM	50.75%	0	0	0
GRU	**63.93%**	0	0	0

Table3. Test Accuracy, Precision, Recall and F1 Score for Different Model

In the overall evaluation of the test dataset, the GRU model shows significantly higher accuracy than the LSTM model. This is mainly because GRU performs better predicting negative samples, resulting in more True Negatives (TN) and fewer False Positives (FP). However, since no positive samples exist in the test dataset, the Precision, Recall, and F1 Score are all zero. This is reflected in the confusion matrix, where both True Positives (TP) and False Negatives (FN) are zero, indicating that the model did not predict any positive instances.

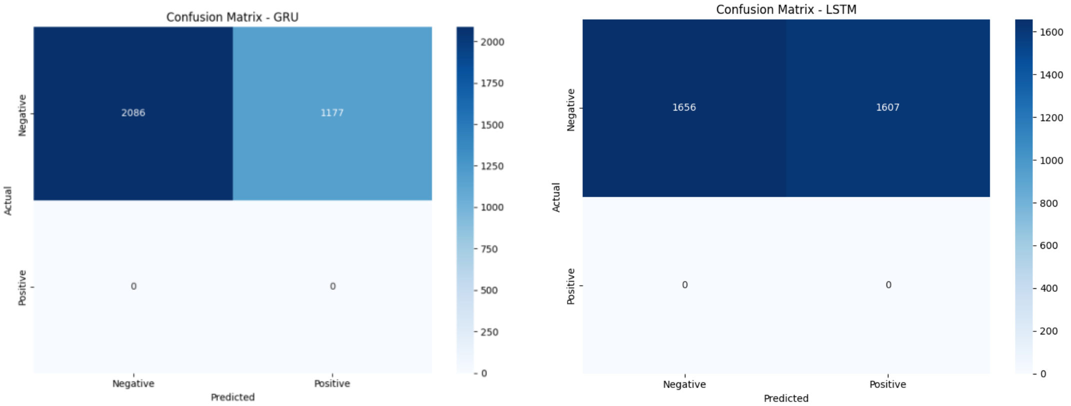

Figure 3 . Confusion Matrixes with different Model

Conclusion

This assignment compared with LSTM and GRU for tweet classification. Results show that the GRU model achieved higher accuracy and precision, trained slightly faster, and used fewer parameters than the LSTM model. GRU demonstrated better overall performance and efficiency, making it a more suitable choice for this task.