Natural Language Processing with Disaster Tweets

Management/Research Question- The research question examines whether a model can automatically determine if a tweet refers to a real disaster. This question is important because emergency response teams, government agencies, and news organizations must process large volumes of social media content during crisis events. An automated classifier allows organizations to identify actionable information more quickly, reduce manual review time, and improve the allocation of resources during emergencies.

Exploratory Data Analysis- Exploratory data analysis was conducted to understand the characteristics of the tweets and identify patterns relevant to modeling.

1. The dataset consists of 7,613 labeled tweets and 3,263 unlabeled test tweets.
2. The classes are moderately imbalanced, with slightly more non-disaster tweets than disaster tweets.
3. Several tweets contain URLs, user mentions, hashtags, and non-alphanumeric characters.
4. Many disaster-related tweets include terms associated with natural events such as “earthquake,” “fire,” “evacuation,” and “emergency,” whereas non-disaster tweets often use similar terms metaphorically.
5. Tweets vary in length, but most contain fewer than 40 tokens, which informed the sequence-length hyperparameter for modeling.

Cross-Validation Design- A cross-validation approach was used to ensure that model performance did not depend on a single training/validation split. The dataset was divided into a training portion and a validation portion using an 80/20 split. The validation set was held constant to ensure comparability across the three RNN architectures.
This approach allowed consistent measurement of accuracy and loss across different hyperparameter settings and model structures.

Model Development- Three recurrent neural network models were created. Each used tokenized and padded text sequences produced by the Keras Tokenizer. Hyperparameters such as embedding dimension, dropout rates, learning rate, and hidden layer size were tuned in order to evaluate differences in performance.

Model 1: Baseline LSTM

- Structure: Embedding layer → LSTM(64) → Dropout(0.3) → Dense(1)
- Validation Accuracy: 0.7827
- This model demonstrated stable learning and produced a strong benchmark score.

Model 2: Bidirectional LSTM

- Structure: Embedding layer → Bidirectional(LSTM(64)) → Dropout(0.4) → Dense(1)
- Validation Accuracy: 0.7590
- Although the model learned temporal patterns from both directions, it showed signs of overfitting after early epochs.

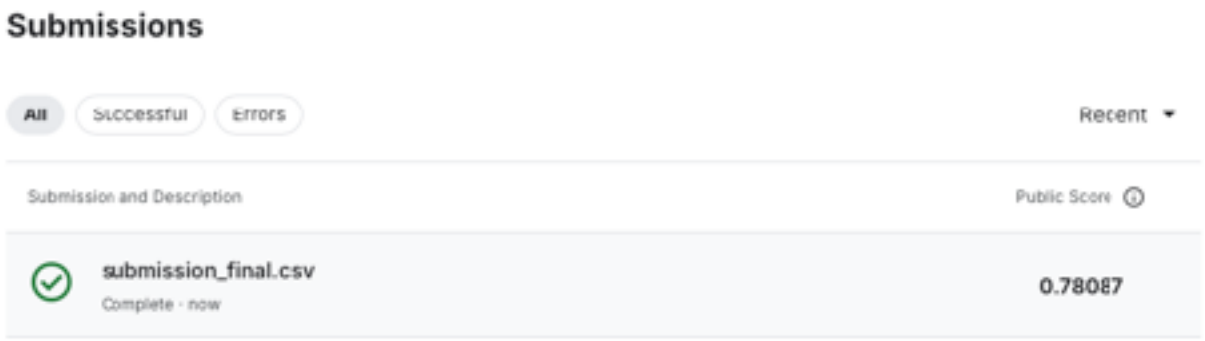
Model 3: GRU

- Initial GRU model stalled during training and produced no meaningful learning.
- A revised GRU model was created with a larger embedding size, increased hidden units, reduced learning rate, and added recurrent dropout.
- Validation Accuracy (improved GRU): approximately 0.74–0.78 depending on the epoch.
- GRU performance was comparable to the LSTM but slightly less stable.

Goodness-of-Fit Metrics- The models were evaluated using accuracy and loss values on the validation set. Across all three models, the LSTM achieved the highest and most consistent validation accuracy.

Model	Training Accuracy	Validation Accuracy	Notes
LSTM	0.9450	0.7827	Best Balance of fit and generalization
BiLSTM	0.9619	0.7590	Overfits after early epochs
Improved GRU	~0.95	~0.74 - ~0.78	Performance comparable after hyper parameter adjustments

Kaggle Submission- Predictions were generated using the best-performing model (LSTM). The predictions were saved in the required format and submitted to the Kaggle competition.



Discussion of Model Performance- The baseline LSTM model provided the strongest generalization among the three architectures. Its validation accuracy and Kaggle score were stable and competitive. The Bidirectional LSTM showed strong training performance but overfitted quickly, which limited its validation accuracy. The GRU model initially failed to learn the classification boundary, which required adjustments to the network size, dropout configuration, learning rate, and batch size. After tuning, the GRU model achieved validation performance similar to the other architectures, although it was somewhat more sensitive to hyperparameter settings. Overall, the results show that recurrent neural networks are capable of identifying disaster-related tweets with reasonable accuracy, and that careful tuning is essential for achieving strong performance on short-text classification tasks.