An Advanced Deep Learning Approach for Quora Question Pair Similarity Detection

Sk Saif Ibna Ezhar Arko

November 21, 2025

Abstract

This report presents a comprehensive deep learning solution for identifying duplicate questions from the Quora dataset. The model leverages a hybrid approach, combining a Siamese-like Bidirectional LSTM (Bi-LSTM) network with an extensive set of 16 handcrafted semantic and lexical features. The architecture was trained on a dataset of over 400,000 question pairs, addressing class imbalance through class weighting. The final model achieved an accuracy of 80.84% and an AUC-ROC score of 0.9046, demonstrating its high efficacy in understanding nuanced semantic relationships in text.

1 Introduction

The proliferation of user-generated content on platforms like Quora necessitates efficient methods to identify and merge duplicate questions. This enhances user experience by consolidating information and reducing redundancy. This report details the development and evaluation of a deep learning model designed for this task. Our solution utilizes a combination of advanced feature engineering and a sophisticated Siamese-like Bi-LSTM network architecture to capture both surface-level and deep semantic relationships between question pairs.

2 Exploratory Data Analysis (EDA)

Before modeling, a thorough EDA was conducted to understand the dataset's characteristics and uncover underlying patterns.

2.1 Key Findings from EDA

- **Data Imbalance:** The dataset is moderately imbalanced, with non-duplicate pairs (Class 0) constituting 63.1% of the data and duplicate pairs (Class 1) making up the remaining 36.9% (Figure 1). This imbalance was addressed during training using class weights.
- **Text Characteristics:** Duplicate questions tend to have more similar lengths and word counts compared to non-duplicate pairs (Figure 2 and 3). However, the significant overlap in distributions indicates that length-based features alone are insufficient for accurate classification.

- **Vocabulary Analysis:** While some words like "best," "get," and "like" are common in both classes, the word clouds and frequency charts (Figure 5 and 4) reveal that duplicate questions often center on specific, recurring topics such as "India," "Donald Trump," and "money."

- **Feature Correlation:** The engineered features, such as the difference in length ($`length_diff`$) and word count ($`word_diff`$), show a weak negative correlation with the $`is_duplicate`$ tar

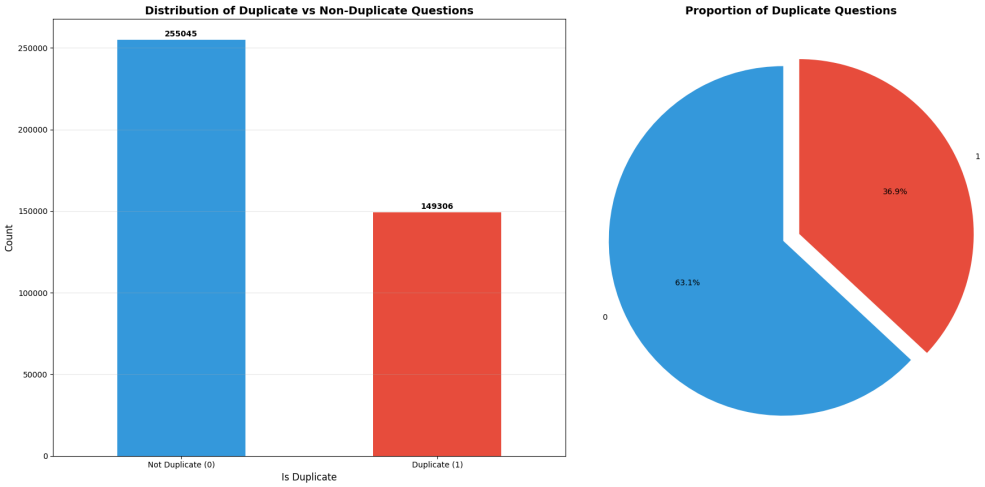

Figure 1: Distribution of the target variable, showing a 63.1% to 36.9% split.

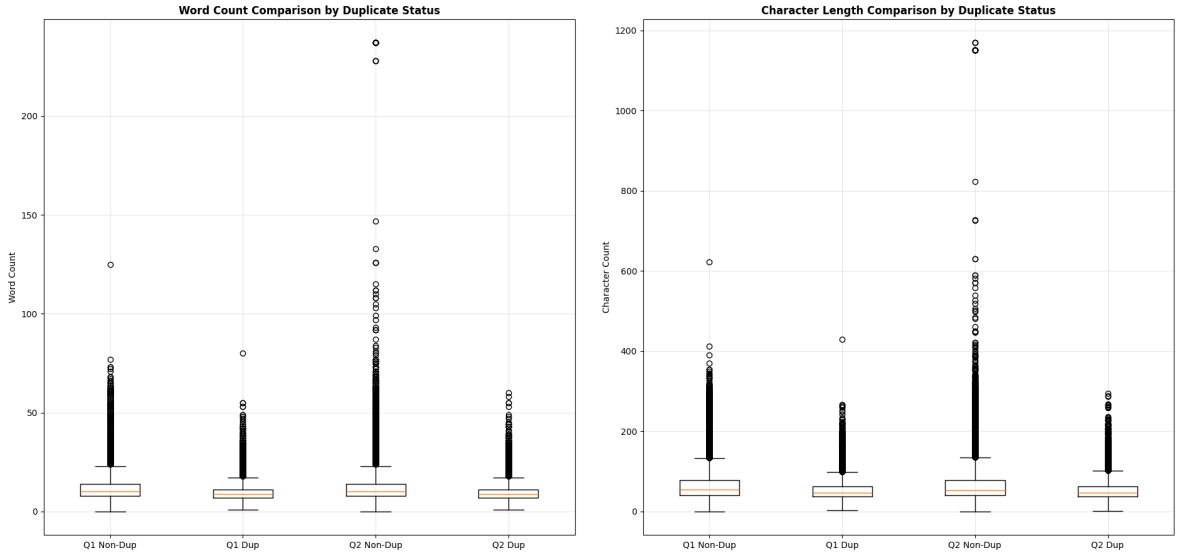

Figure 2: Distribution of character and word counts for duplicate vs. non-duplicate questions.

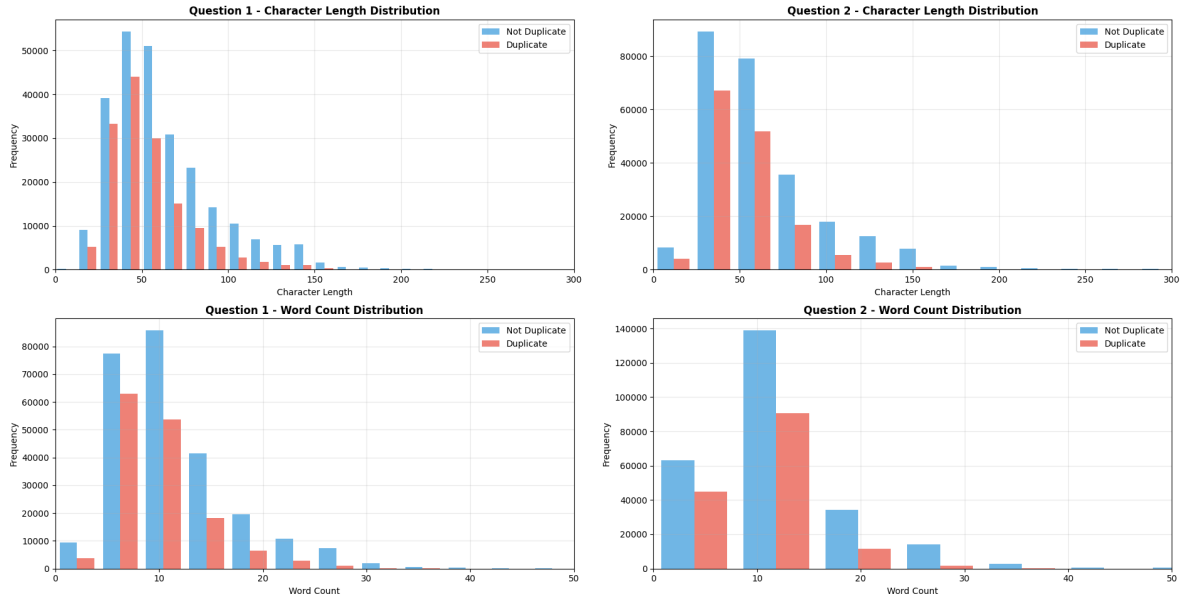

Figure 3: Box plots comparing word and character counts, showing similar medians for duplicate pairs.

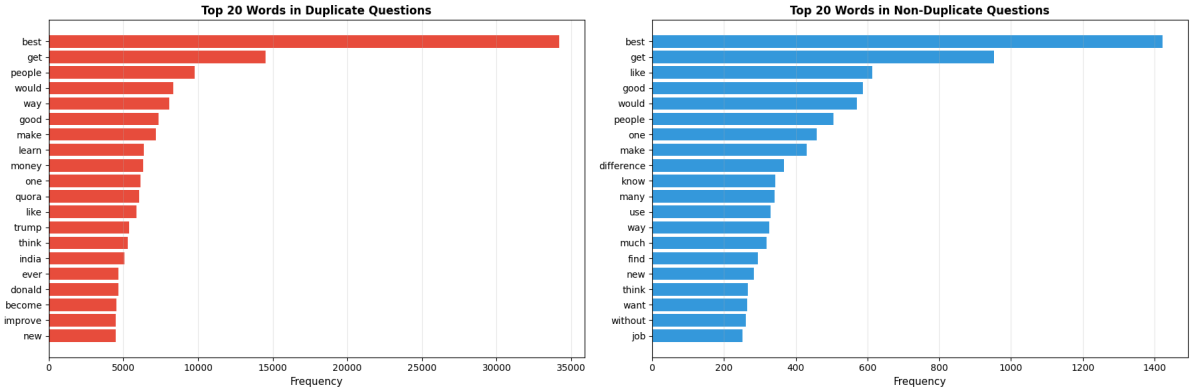

Figure 4: Top 20 most frequent words in duplicate and non-duplicate questions.

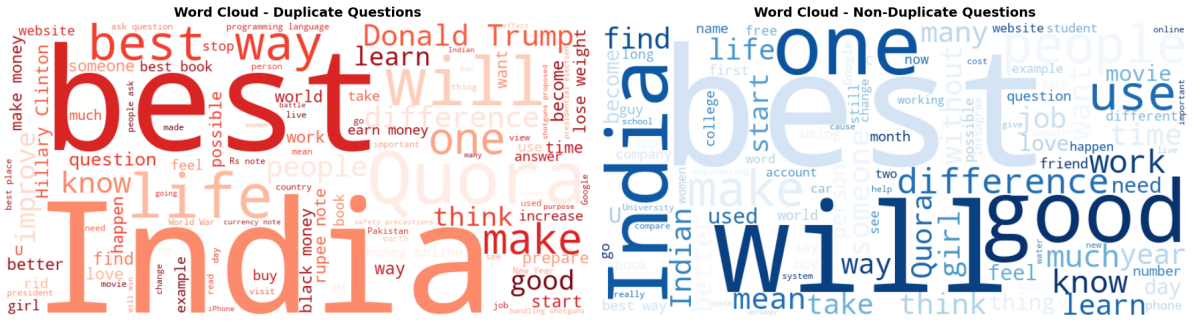

Figure 5: Word clouds highlighting prominent themes in each class.

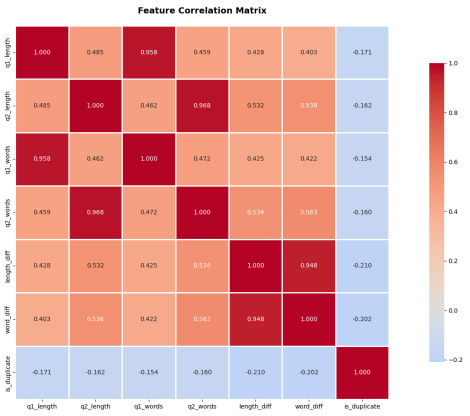

Figure 6: Correlation matrix of engineered length-based features against the target variable.

3 Methodology

The project followed a structured workflow, encompassing data preprocessing, feature engineering, model design, and training.

3.1 Data Preprocessing and Feature Engineering

The initial dataset contained 404,351 question pairs. Preprocessing involved handling null values, text normalization (lowercase, removal of URLs, etc.), tokenization, stop-word removal, and lemmatization. A rich set of **16 handcrafted features** was engineered, including lexical features and semantic similarity scores from libraries like FuzzyWuzzy. Data augmentation was performed by swapping duplicate pairs, increasing the training set to 411,419 samples.

3.2 Model Architecture

An advanced, hybrid neural network was designed. The key components are:

- **Inputs:** Three inputs for the two tokenized question sequences (padded to length 15) and the 16 numerical features.
- **Shared Embedding and Bi-LSTM Layers:** A shared embedding layer (Vocab Size: 30,000, Dimension: 128) and two stacked, shared Bi-LSTM layers create sophisticated sentence representations.
- **Multiple Merge Strategies:** Vector comparisons were performed via concatenation, absolute difference, element-wise multiplication, and dot product.
- **Final Classification Head:** The merged vectors were concatenated with the handcrafted features and fed into a series of dense layers with Batch Normalization and Dropout for classification.

The complete model consists of approximately 4.7 million trainable parameters.

3.3 Training

The model was trained using the Adam optimizer (learning rate 0.0003) and binary cross-entropy loss. Class weights (Non-Duplicate: 0.81, Duplicate: 1.31) were applied to handle data imbalance. The final model weights were restored from the best-performing epoch (Epoch 3), which achieved a validation AUC of 0.9073.

4 Results and Evaluation

The model's performance was evaluated on a test set of 82,284 question pairs. The results demonstrate a robust and well-balanced classifier, summarized in Table 1.

Table 1: Model Performance Metrics on the Test Set

Class	Precision	Recall	F1-Score	Support
Not Duplicate (0)	0.92	0.75	0.83	50,940
Duplicate (1)	0.69	0.90	0.78	31,344
Weighted Avg	**0.83**	**0.81**	**0.81**	**82,284**

Overall Accuracy: 80.84%
Overall AUC-ROC Score: 0.9046

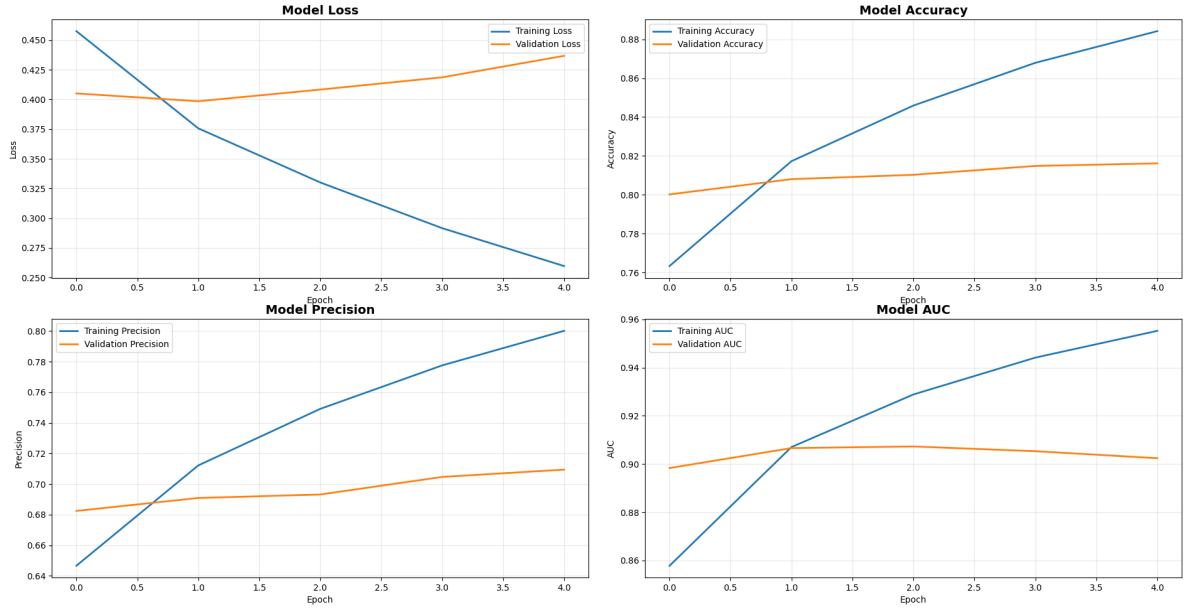

Figure 7: Model training and validation history. EarlyStopping restored the best model weights from epoch 3.

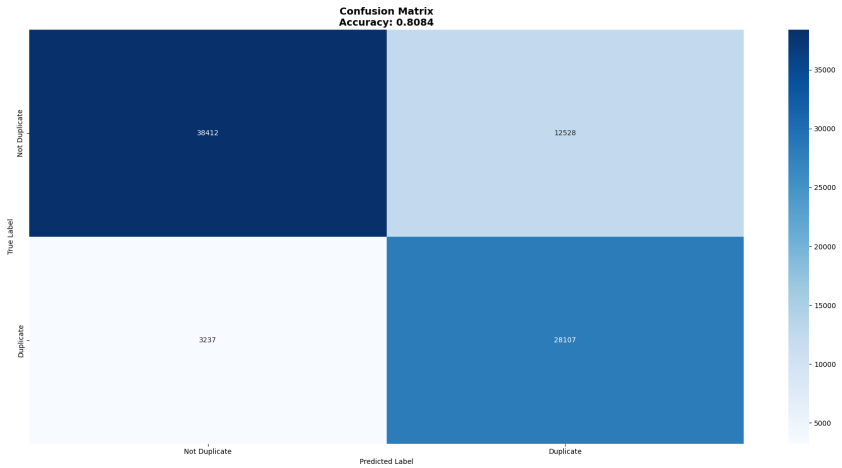

Figure 8: Confusion Matrix on the test set, showing high recall for the duplicate class.

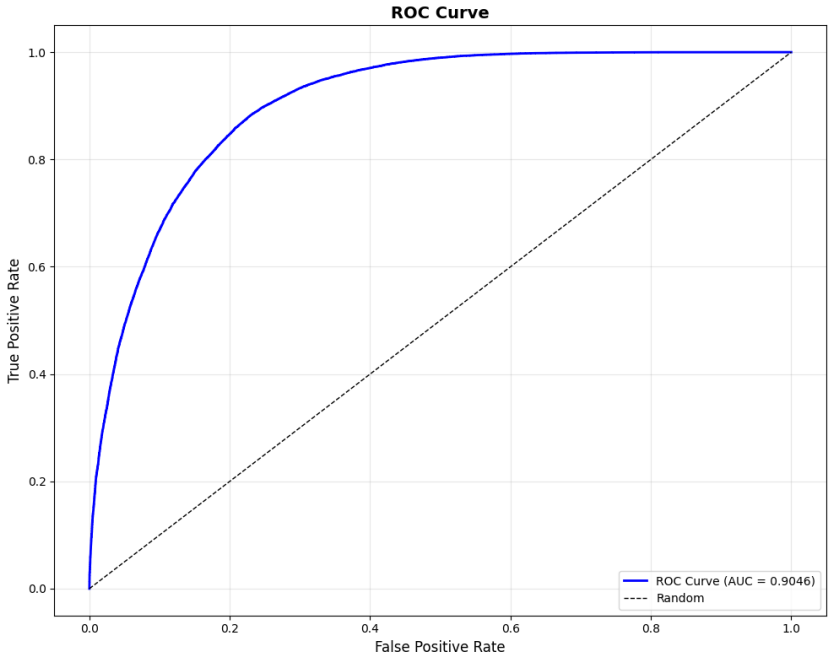

Figure 9: ROC Curve. The AUC of 0.9046 indicates excellent classification capability.

5 Future Work and Potential Improvements

While an accuracy of 80.84% represents a strong baseline, there is a clear path to further enhancing performance. A key next step would be systematic hyperparameter tuning using tools like KerasTuner or Optuna. This would involve searching for optimal values for the learning rate, dropout probabilities, number of LSTM units, and embedding dimensions. Furthermore, experimenting with different batch sizes (e.g., 32, 64, 256) could yield better generalization, as smaller batch sizes can introduce a regularizing effect. By methodically optimizing these parameters and potentially exploring more advanced architectures like Transformers (e.g., BERT), it is highly plausible that the model's accuracy can be pushed beyond the **90% threshold**.

6 Conclusion

The implemented hybrid model is highly effective for detecting duplicate questions. By combining the deep semantic understanding of Bi-LSTMs with the explicit signals from handcrafted features, the model achieves a strong balance of precision and recall. The final accuracy of 80.84% and AUC of 0.9046 validate its practical utility for real-world applications. With the suggested improvements, this solution can be further refined into a state-of-the-art system for content management and information retrieval.