IDENTIFICATION OF QUORA QUESTION PAIRS

CAPSTONE PROJECT REPORT

Submitted by

PG ADSML - Cohort 4’s Group 1 Members

Akhil AD
Lakshmipriya
Nitin Shashi
Prabu Balamurugan
Sakhinala Venkata Prabhath
Sakthe Priya P
Sathish Kumar Selvaraj
Saumya Tripathi
Vivek Babu B

IDENTIFICATION OF QUORA QUESTION PAIRS

Contents

1. Introduction

1.1 Abstract:

This project aims to improve the user experience on Quora by identifying duplicate question pairs. Leveraging Natural Language Processing (NLP) and Deep Learning (DL) algorithms, we classify whether questions share the same intent. With a dataset of over 400,000 labeled question pairs, we train and evaluate models to optimize the process of seeking and providing high-quality answers on Quora.

1.2 Objective:

- To utilize NLP and DL algorithms to extract semantic meaning from question pairs.
- To develop a model capable of classifying if pairs of questions on Quora have the same intent are duplicate.
- To test the data across multiple models and identify the effective model.
- To optimize the model's performance to achieve a high accuracy in identifying duplicate question pairs.

1.3 Metrics:

The confusion matrix helps assess classification model performance in machine learning by comparing predicted values against actual values for a dataset[1]. Here are the key components of a confusion matrix:

1. **True Positives (TP)**: Number of correct predictions for the positive class
2. **True Negatives (TN)**: Number of correct predictions for the negative class
3. **False Positives (FP)**: Negative-class instances incorrectly identified as positive
4. **False Negatives (FN)**: Actual positive instances incorrectly identified as negative

The confusion matrix helps us understand the following metrics:

- **Accuracy:** The ratio of total correct instances to the total instances
  $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$
- **Precision:** True positive predictions among all positive predictions
  $Precision = \frac{TP}{TP + FP}$
- **Recall (Sensitivity):** True positive predictions among all actual positive instances
  $Recall = \frac{TP}{TP + FN}$
- **F1 Score:** Balances precision and recall
  $F1\ score = \frac{2*Precision*Recall}{Precision + Recall}$
- **Specificity:** is the true negative rate – proportion of true negative predictions among all actual negative instances.
  $Specificity = \frac{TN}{TN+ FP}$

2. Analysis

2.1 Data Exploration:

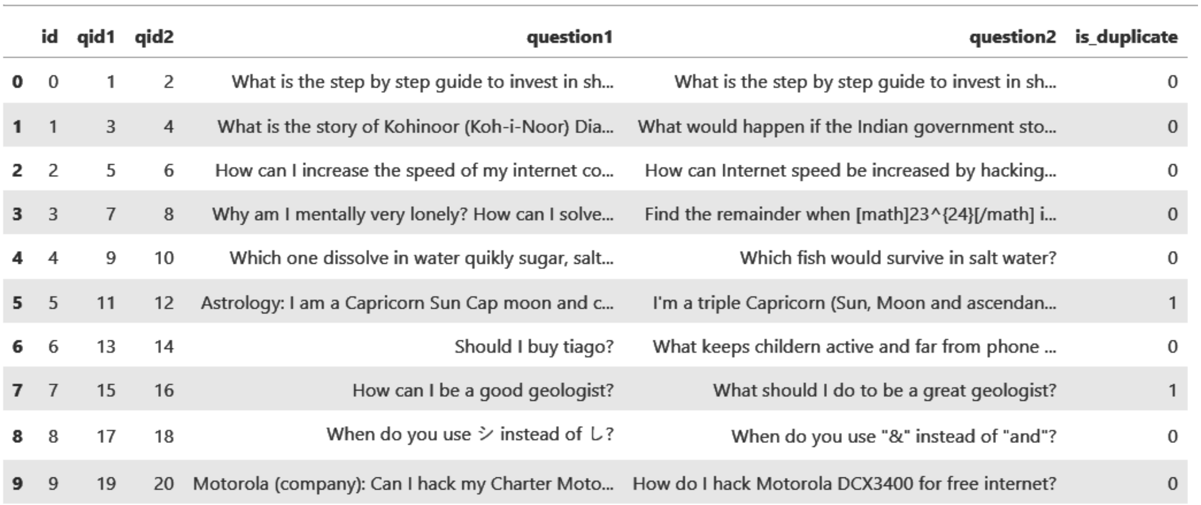
Quora question dataset provided on the Kaggle Competition (approx. 4 Million records) : **https://www.kaggle.com/quora/question-pairs-dataset**

The dataset contains 404351 rows and 6 columns (id, qid1, qid2, question1, question1, is_duplicate) Features:

- **id** - the id of a training set question pair (Datatype: Int64)
- **qid1, qid2** - unique ids of each question (Datatype: Int64)
- **question1, question2** - the full text of each question (Datatype: String)
- **is_duplicate** - the target variable, duplicate question pairs or questions with similar meaning are represented by 1 and non duplicate pairs are denoted by 0 (Datatype: Int64)

Fig01: Top 10 rows of the dataset along with the columns names

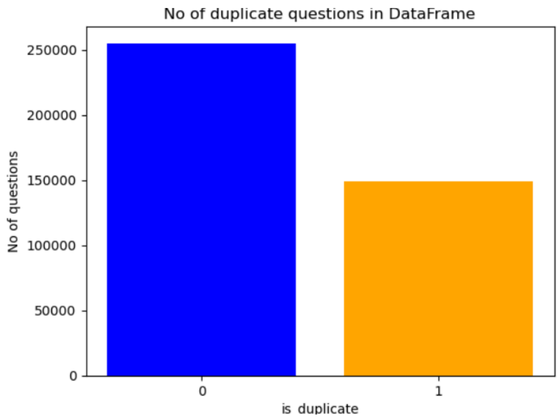
1.

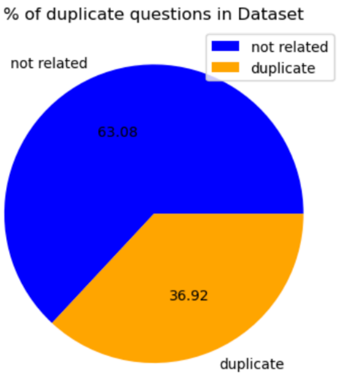


	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0
5	5	11	12	Astrology: I am a Capricorn Sun Cap moon and c...	I'm a triple Capricorn (Sun, Moon and ascendan...	1
6	6	13	14	Should I buy tiago?	What keeps childern active and far from phone ...	0
7	7	15	16	How can I be a good geologist?	What should I do to be a great geologist?	1
8	8	17	18	When do you use シ instead of し?	When do you use "&" instead of "and"?	0
9	9	19	20	Motorola (company): Can I hack my Charter Moto...	How do I hack Motorola DCX3400 for free internet?	0

Observations related to the dataset:

- Total number of duplicate questions in the dataset: 1,49,306
- Percentage of duplicate questions in the dataset: 36.92%





2.1 Exploratory Visualization:

For exploratory data visualization, word cloud and tf idf share between questions gives us a fair idea about the similarities between terms or questions.

- **Word cloud** is one of the most powerful visualization methods when it comes to text data. The size of words is dependent on the occurrence frequency. Here we have generated 2 separate plots using word cloud library for both question1 and question2

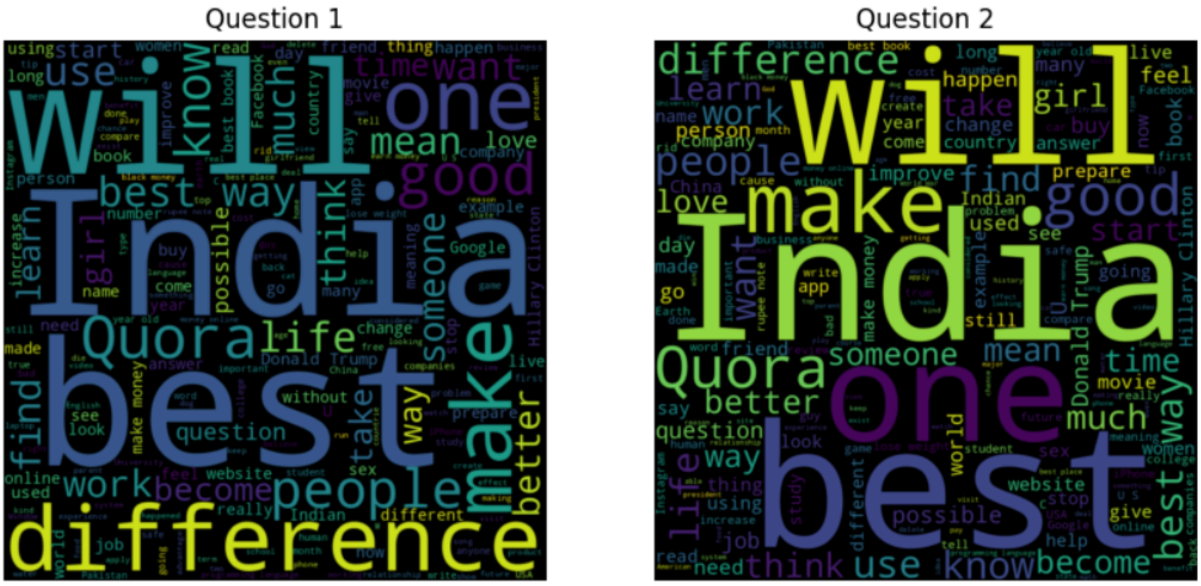
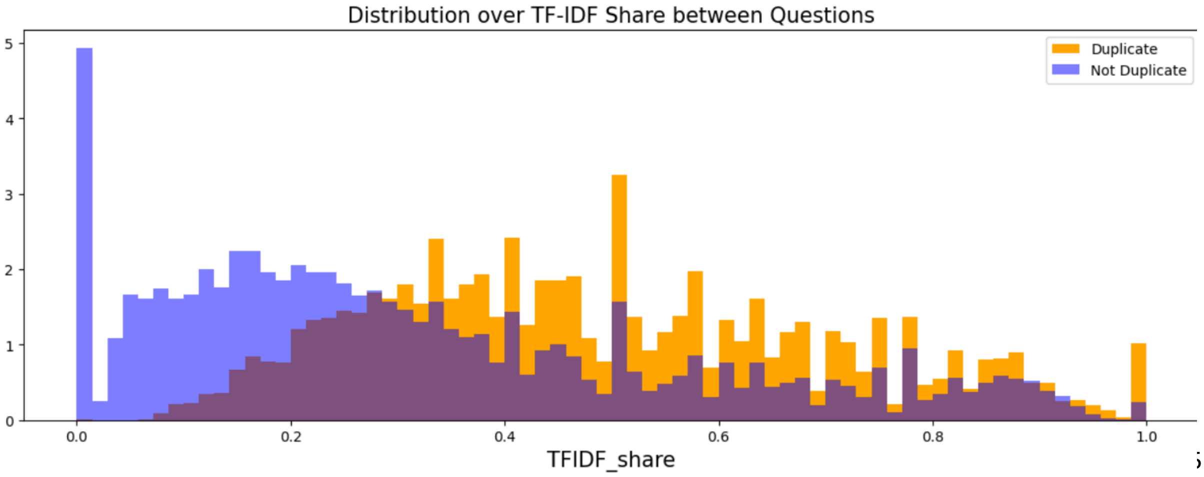


Fig02: Word Cloud of Question 1 & Question 2

The above two plots clearly shows that there are some frequently occuring common words in the two questions (question1 and question2) and have similar frequency. Ex: India, will, best, good etc.

- **TF-IDF wordshare** between questions is a measure of similarity between two text documents. TF-IDF word share provides a numerical value indicating the degree of similarity between two questions based on the importance of shared words as measured by their TF-IDF scores.

  We can clearly see that there is intersection of tf-idf for duplicates and non-duplicate question pairs. That infers there are similar terms in both the sets.



2.3 Algorithms and Techniques

Supervised learning algorithms are used for this binary classification task (duplicate or non_duplicate)

Following Supervised/Deep Learning Algorithms were implemented:

1. ***Logistic Regression***: Logistic regression is a data analysis technique that uses mathematics to find the relationships between two data factors. It then uses this relationship to predict the value of one of those factors based on the other. The prediction usually has a finite number of outcomes, like yes or no. It computes a weighted sum of input features and outputs the logistic of the results. It is simple, fast and easy to interpret. It has low variance, so is less prone to overfitting.

2. ***Support Vector Machine***: A support vector machine (SVM) is a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane that maximizes the distance between each class in an N-dimensional space. It analyzes both classification and regression tasks.

3. ***Random forest***: A Random Forest Algorithm is a supervised machine learning algorithm that is extremely popular and is used for Classification and Regression problems in Machine Learning. The greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability. Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model. This combination of multiple models is called Ensemble.

   Ensemble uses two methods:
   Bagging: Creating a different training subset from sample training data with replacement is called Bagging. The final output is based on majority voting.
   Boosting: Combining weak learners into strong learners by creating sequential models such that the final model has the highest accuracy is called Boosting.

   This algorithm reduces overfitting and thus, gives high accuracy. It automatically learns feature interactions and output feature importance. It is very expensive and very difficult to interpret.

4. ***LSTM:*** Long Short-Term Memory Networks is a deep learning, sequential neural network that allows information to persist. It is a special type of Recurrent Neural Network which is capable of handling the vanishing gradient problem faced by RNN. LSTM resolves the problem caused by traditional RNNs and machine learning algorithms. The LSTM Model can be implemented in Python using the Keras library.

   This Deep Learning algorithm is a specialized form of recurrent neural network (RNN) designed to overcome the limitations of its predecessors. Its ability to capture long-term dependencies in sequential data makes it invaluable for applications ranging from natural language processing to time series forecasting and anomaly detection.

   In practical implementations, Python's Keras library provides a user-friendly interface for building LSTM models. With Keras, developers can easily construct, train, and evaluate LSTM networks for a wide range of tasks. Its intuitive API allows for rapid prototyping and experimentation, making it a popular choice among data scientists and machine learning practitioners.

   Moreover, the flexibility of LSTM networks extends beyond traditional classification tasks. By leveraging their tree-like structure, LSTM models can be adapted for binary classification at each node, facilitating more complex decision-making processes. This capability opens up avenues for applications such as hierarchical classification and structured prediction, where decisions are made at multiple levels of abstraction.

   In summary, LSTM networks represent a significant milestone in the field of deep learning, offering a robust solution for modeling sequential data and addressing the challenges posed by traditional RNNs. With their ability to preserve information over long sequences, coupled with the ease of implementation using libraries like Keras, LSTM networks are poised to remain at the forefront of machine learning research and application development.

3. Methodology

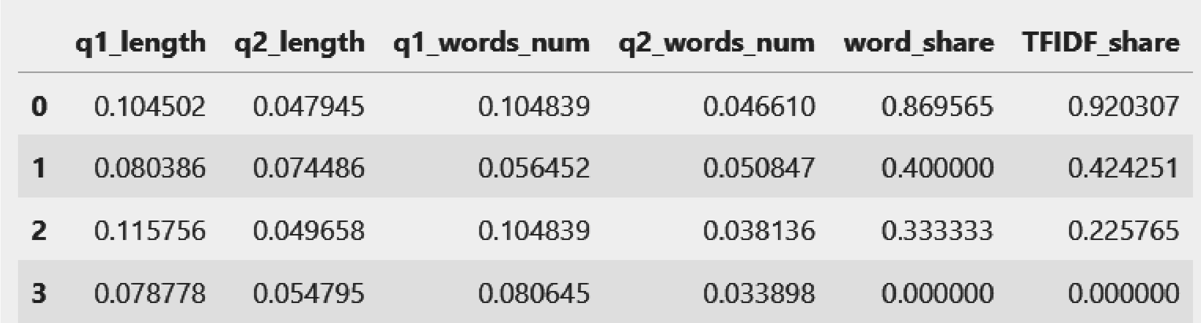
3.1 Data Preprocessing

Before the dataset is given to the model for training, pre-processing needs to be performed.

1. **Data cleaning & Feature Selection:** Retaining only Non-Null rows in the dataset and removing unwanted columns which does not contribute to the training model. Here we removed id, qid1, qid2 represents the serial numbers which does not have any valuable information regarding the meaning or pattern in the questions

2. **Feature Extraction:** We have constructed the following features from the dataset:
   1. **Length of the questions** (Q1 & Q2): Entire String length of each questions
   2. **Number of words** (Q1 & Q2): Word count of each questions
   3. **Word share**: The number of words in both question1 and question2 divided by the total number of words in question1 and question2
   4. **TF-IDF word share**: For finding the tf-idf word share, CountVectorizer and TfidfTransformer from sklearn were used; which converts a collection of raw documents to a matrix of TF-IDF features

3. **Feature Scaling:** Normalization is done to standardize the range of numerical features. It standardizes features by removing the mean and scaling to unit variance. Here we have used **MIn-Max Scalar**. It scales numerical features to a specified range, typically between 0 and 1. It operates on each feature independently and transforms it such that the minimum value of the feature becomes 0, and the maximum value becomes 1, while preserving the relative distance between data points, improving the convergence and performance of machine learning models.



	q1_length	q2_length	q1_words_num	q2_words_num	word_share	TFIDF_share
0	0.104502	0.047945	0.104839	0.046610	0.869565	0.920307
1	0.080386	0.074486	0.056452	0.050847	0.400000	0.424251
2	0.115756	0.049658	0.104839	0.038136	0.333333	0.225765
3	0.078778	0.054795	0.080645	0.033898	0.000000	0.000000

Fig04: Dataframe after preprocessing; Top 4 rows along with the columns names

3.2 Implementation

After data preprocessing, we split the data set to train and test (80-20% split). We have trained our dataset on 4 models. For each model, we have varied respective parameters and plot the model performance. From the plot we get the best value of the parameter of each model, resulting in higher prediction and less overfitting.

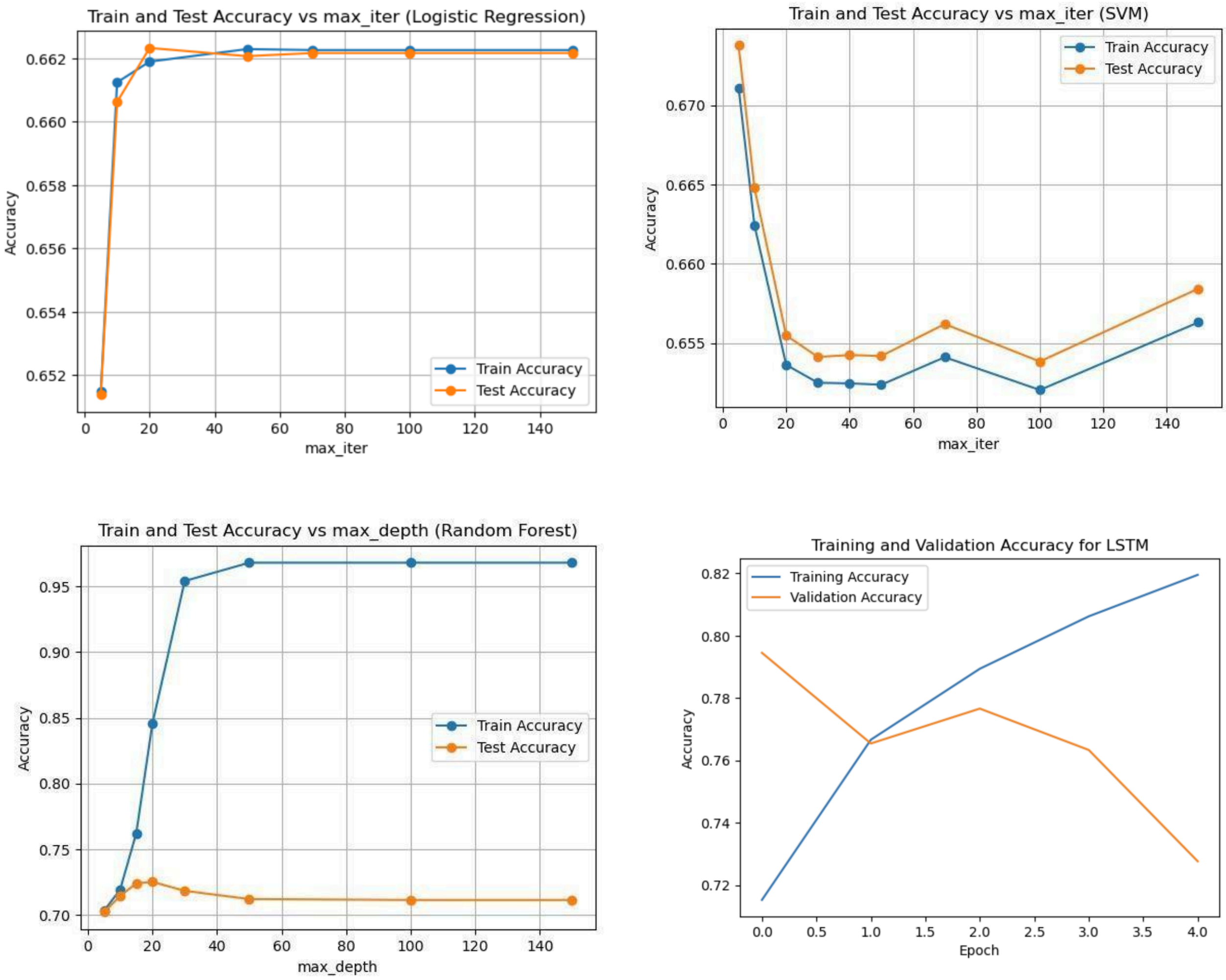
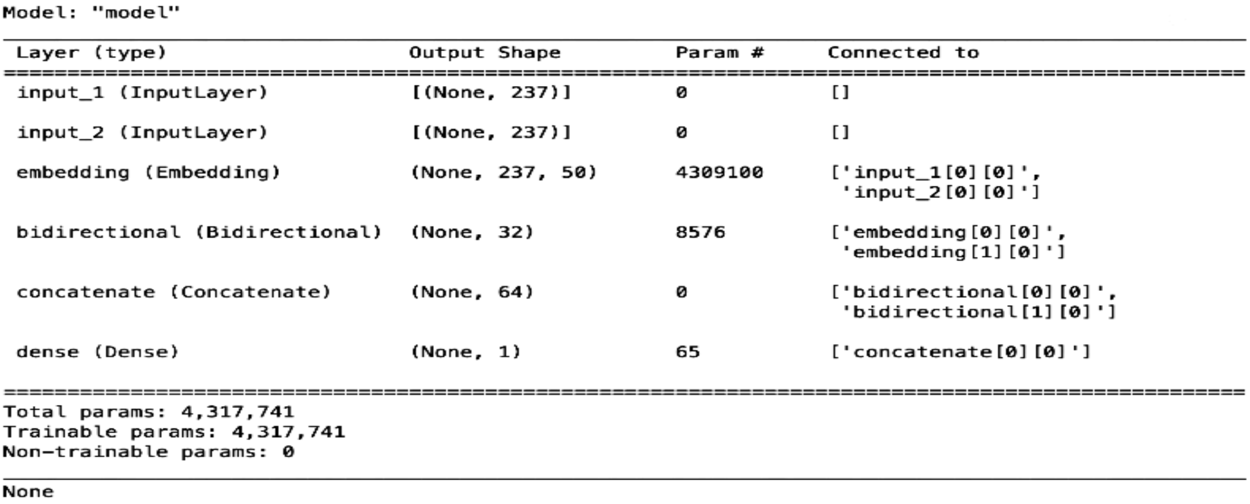


Fig05: Plots of all the 4 model's performance with change in respective parameter

LSTM Model :



```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 237)]        0           []

 input_2 (InputLayer)           [(None, 237)]        0           []

 embedding (Embedding)          (None, 237, 50)      4309100     ['input_1[0][0]',
                                                                  'input_2[0][0]']

 bidirectional (Bidirectional)  (None, 32)           8576        ['embedding[0][0]',
                                                                  'embedding[1][0]']

 concatenate (Concatenate)      (None, 64)           0           ['bidirectional[0][0]',
                                                                  'bidirectional[1][0]']

 dense (Dense)                  (None, 1)            65          ['concatenate[0][0]']

==================================================================================================
Total params: 4,317,741
Trainable params: 4,317,741
Non-trainable params: 0
__________________________________________________________________________________________________
None
```

This model is designed for text input data represented by sequences of tokens. It consists of two input layers, each accepting sequences of equal length. These sequences are then passed through an embedding layer, which maps each token to a dense vector representation. The embedded sequences are then fed into a bidirectional LSTM layer with 16 units, utilizing dropout for regularization. The outputs of the two LSTMs are concatenated and passed through a dense layer with a sigmoid activation function, yielding a binary classification output. The model is trained using binary cross-entropy loss and optimized with the Adam optimizer, aiming to maximize classification accuracy.

3.3 Hyperparameter Tuning

From the above plots, and also from the metrics (Precision, F1 score) we can select the values of the parameters for each model having best performance and use the respective parameters for training our model.

Table01: Following are the selected values of the parameters for each model:

Model	**Logistic Regression**	**SVM**	**Random Forest**	**LSTM**

Parameters	max_iter=**50**	max_iter=**5**	max_depth=**15**	epochs = **5**
				drop_out = **0.2**

4. Results

After hyperparameter tuning, we get the following model performance:

Table02: Test and Train accuracy of respective models

Model	**Logistic Regression**	**SVM**	**Random Forest**	**LSTM**
Train Accuracy	66.22%	67.11%	76.23%	83.10%
Test Accuracy	66.20%	67.37%	72.41%	82.40%

Table03: Model performance on various metrics

Model	**Logistic Regression**	**SVM**	**Random Forest**	**LSTM**
Confusion Matrix	[[40036 10973] [16355 13506]]	[[33360 17649] [8732 21129]]	[[36602 14407] [7902 21959]]	[[164286 39750] [17197 102247]]
Precision	0.551	0.544	0.603	0.7144

4.1 Prediction on Existing Data:

Prediction question pairs	**Model Prediction Performance**
Question1: What can make Physics easy to learn? **Question2:** How can you make physics easy to learn? **y_true = 1**	Prediction for: LogisticRegression [0.70695481] **Duplicate** Prediction for: SVC [0.24943712] **Non-duplicate** Prediction for: RandomForestClassifier [0.61790359] **Duplicate** Prediction for: LSTM [0.8244267] **Duplicate**
Question1: What is the best travel website in spain? **Question2:** What is the best travel website? **y_true =0**	Prediction for: LogisticRegression [0.53452218] **Duplicate** Prediction for: SVC [0.2638796] **Non-duplicate** Prediction for: RandomForestClassifier [0.49316933] **Non-duplicate** Prediction for: LSTM [0.50174993] **Duplicate**
Question1: Should I repeat 2nd year in college, or find a new college? It's a 5 year course. **Question2:** I study in Ashutosh College in Kolkata. Can I study in any other college in India in my 2nd year?	Prediction for: LogisticRegression [0.09687553] **Non-duplicate** Prediction for: SVC [0.40670622] **Non-duplicate**

y_true = 0	Prediction for: RandomForestClassifier [0.02148022] **Non-duplicate** Prediction for: LSTM [0.057] **Non-duplicate**

4.2 Prediction on New Data:

Question1: what are the best places to visit in Chennai? **Question2:** list famous places in Chennai during vacation	Prediction for: LogisticRegression [0.24330351] **Non-duplicate** Prediction for: SVC [0.43678635] **Non-duplicate** Prediction for: RandomForestClassifier [0.25787164] **Non-duplicate** Prediction for: LSTM [0.38491333] **Non-duplicate**
Question1: Why is Sachin Tendulkar known as God of Cricket? **Question2:** Why Sachin Tendulkar is the best cricketer in the world	Prediction for: LogisticRegression [0.45911128] **Non-duplicate** Prediction for: SVC [0.36395852] **Non-duplicate** Prediction for: RandomForestClassifier [0.52530384] **Duplicate** Prediction for: LSTM [0.50741625] **Duplicate**
Question1: Which iphone model should I buy in 2024 **Question2:** Should I buy Samsung or Iphone?	Prediction for: LogisticRegression [0.42152845] **Non-duplicate** Prediction for: SVC [0.31838029]

	Non-duplicate Prediction for: RandomForestClassifier [0.41671605] **Non-duplicate** Prediction for: LSTM [0.450834] **Non-duplicate**

5. Conclusion

In conclusion, the analysis of Quora question pairs using LSTM and Random Forest models has shown promising results, with LSTM exhibiting superior accuracy compared to other models. However, it's noteworthy that SVM performed poorly in certain test examples when compared to Logistic Regression. This suggests that while LSTM and Random Forest are effective for this task, the choice of algorithm should be carefully considered based on the specific characteristics of the data.

Furthermore, this solution can be extended to other areas where text similarity or question pair analysis is crucial. For instance, in customer support systems, this approach can help in identifying similar queries and providing relevant responses efficiently. Similarly, in content moderation platforms, it can assist in flagging potentially duplicate or redundant content. The versatility of these models opens up opportunities for their application across various domains requiring text analysis and similarity assessment.

6. References

1. https://www.ibm.com/topics/confusion-matrix
2. https://aws.amazon.com/what-is/logistic-regression/
3. https://www.ibm.com/topics/support-vector-machine
4. https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm
5. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/