MS in Data Science Program
Statistical Software
Instructor: Dr. Ajita John

RUTGERS

16:954:577:01 STATISTICAL SOFTWARE

Project Report

"QUORA QUESTION PAIR DETECTION"

TEAM 7

ASHISH N KADAM	**ANK120**
NEERAJ CHAUDHARI	**NC912**
FNU VASUREDDY	**FV121**

TEACHING ASSISTANTS: **WEI YUAN** AND **YUNJIAO BAI**

SEMESTER**: FALL 2024**

"The goal of NLP is to make computers understand human language, not just recognize words, but grasp meaning."

- *Andrew Ng*

Contents

1 | Introduction

Quora is a widely used question-and-answer platform that allows users to ask questions on a variety of topics and receive answers from other community members. Launched in 2009, Quora has grown into one of the largest knowledge-sharing platforms globally, with millions of active users contributing to a diverse range of content. The platform's goal is to create a comprehensive, user-driven knowledge base where individuals can seek and share information across subjects ranging from technology and business to personal experiences and niche hobbies. Quora's unique feature is its community-driven content model, where users can upvote, comment, and edit answers, making it a dynamic space for collaborative learning and idea exchange.

Quora, as one of the most popular question-and-answer platforms, faces a significant challenge in managing the vast amount of content being continuously posted by its users. A common issue that arises is the existence of duplicate questions—where multiple users post nearly identical or very similar questions. This not only clutters the platform but also leads to inefficiencies in the system, as users may not always find the most relevant answers when searching for solutions.

2 | Problem Statement

Despite existing mechanisms for question search and categorization, users often face difficulties in identifying whether their question has already been asked and answered. Additionally, search results may return questions that are semantically similar but not exact matches, leading to reduced accuracy in the relevance of results.To address this problem, the aim of this project is to develop a robust model that can identify already answered questions on Quora by evaluating the semantic similarity between new and existing questions. By leveraging advanced Natural Language Processing (NLP) techniques, such as sentence transformers, the model will be trained to measure the semantic similarity between questions more accurately. This would help users avoid reposting duplicate questions, improve the efficiency of the search function, and ultimately enhance the overall user experience by delivering more relevant content.

3 | Dataset

The dataset utilized for this project is publicly available on Kaggle and can be accessed through the following link: **https://www.kaggle.com/datasets/quora/question-pairs-dataset**. This dataset is an extensive collection designed to facilitate the identification of duplicate question pairs, and it consists of a total of 400,000 individual records. Each record in the dataset represents a pair of questions, along with additional features that provide useful metadata for analysis.

The dataset is structured into six distinct attributes or columns, each serving a unique role in the analysis. These attributes are designed to provide information about the questions in the pairs, as well as a binary label indicating whether the two questions in the pair are duplicates.The six attributes are outlined in detail below:

1. **id**: A unique identifier for each record in the dataset. This column helps to differentiate between different question pairs.
2. **qid**: An identifier for the first question in the pair. This helps to group questions that are related or belong to the same category.
3. **qid2**: An identifier for the second question in the pair. This column, in conjunction with qid, allows us to track which questions are being compared in each record.
4. **question1**: The text of the first question in the pair. This is the main content that will be analyzed for similarity with the second question.
5. **question2**: The text of the second question in the pair. This question will be compared with the first question to determine if they are semantically similar or not.
6. **is_duplicate**: A binary target variable that indicates whether the two questions in the pair are duplicates of each other. A value of 1 signifies that the questions are duplicates, while a value of 0 means they are not.

ID	QID1	QID2	QUESTION1	QUESTION2	IS_DUPLICATE
0	1	2	What is the step by step guide to invest in share market in india?	What is the step by step guide to invest in share market?	0
1	3	4	What is the story of Kohinoor (Koh-i-Noor) Diamond?	What would happen if the Indian government stole the Kohinoor (Koh-i-Noor) diamond back?	0
2	5	6	How can I increase the speed of my internet connection while using a VPN?	How can Internet speed be increased by hacking through DNS?	0
3	7	8	Why am I mentally very lonely? How can I solve it?	Find the remainder when [math]23^{24}[/math] is divided by 24,23?	0
4	9	10	Which one dissolve in water quikly sugar, salt, methane and carbon di oxide?	Which fish would survive in salt water?	0
5	11	12	Astrology: I am a Capricorn Sun Cap moon and cap rising...what does that say about me?	I'm a triple Capricorn (Sun, Moon and ascendant in Capricorn) What does this say about me?	1
6	13	14	Should I buy tiago?	What keeps childern active and far from phone and video games?	0
7	15	16	How can I be a good geologist?	What should I do to be a great geologist?	1
8	17	18	When do you use ã‚· instead of ã—?	When do you use "&" instead of "and"?	0
9	19	20	Motorola (company): Can I hack my Charter Motorolla DCX3400?	How do I hack Motorola DCX3400 for free internet?	0
10	21	22	Method to find separation of slits using fresnel biprism?	What are some of the things technicians can tell about the durability and reliability of Laptops and its components?	0
11	23	24	How do I read and find my YouTube comments?	How can I see all my Youtube comments?	1
12	25	26	What can make Physics easy to learn?	How can you make physics easy to learn?	1
13	27	28	What was your first sexual experience like?	What was your first sexual experience?	1
14	29	30	What are the laws to change your status from a student visa to a green card in the US, how do they compare to the immigration laws in Canada?	What are the laws to change your status from a student visa to a green card in the US? How do they compare to the immigration laws in Japan?	0
15	31	32	What would a Trump presidency mean for current international masterâ€™s students on an F1 visa?	How will a Trump presidency affect the students presently in US or planning to study in US?	1
16	33	34	What does manipulation mean?	What does manipulation means?	1

Figure 3.1: Input Data Schema

4 | Exploratory Analysis, Data Preparation, and Feature Engineering

4.1 | EDA and Data Preparation

Exploratory Data Analysis (EDA) is a crucial and foundational step in any data science project. It serves as the first stage of data analysis, allowing us to understand the underlying structure of the data, identify patterns, detect outliers, and uncover relationships among features. Before proceeding with any modeling or advanced techniques, it is essential to thoroughly explore and preprocess the dataset.

By performing EDA, we gain insights into the data's characteristics, distribution, and potential issues that could affect the outcomes of subsequent analyses. This process helps in identifying important trends and anomalies that guide the selection of appropriate models, feature engineering strategies, and evaluation methods. The goal of EDA is not only to prepare the data for further analysis but also to establish a clear understanding of its behavior, which plays a key role in making informed decisions throughout the project.

To ensure that the data is well-prepared for modeling, a series of EDA steps are carried out, such as examining summary statistics, visualizing data distributions, identifying missing or erroneous values, and exploring correlations between features. These steps provide essential insights that are critical for making key decisions and formulating strategies for the next stages of the project.The basic steps of EDA and Data Prep are as follows:

1. **Null Values Handling**: Checking the dataset for any missing or null values and deciding whether to remove or impute them based on the nature of the data and the impact on the analysis.
2. **Duplicate Handling**: Identifying and handling any duplicate records in the dataset, ensuring that only unique entries are considered for further analysis to prevent biases in model training.
3. **Text Cleaning in the Questions**: Processing the text data by removing unnecessary characters, special symbols, and correcting any formatting issues to standardize the questions for analysis and modeling.
4. **Sentence Length Distribution**: Analyzing the distribution of sentence lengths across all questions to understand the overall structure and identify any outliers or inconsistencies in the data.

5. **Word Cloud Visualization**: Visualizing the most frequent words or terms in the dataset using a word cloud, providing insights into the common themes and topics discussed in the questions.

4.2 | Feature Engineering

Feature engineering is the process of selecting, modifying, or creating new features from raw data to improve the performance of machine learning models. It is a crucial step because well-engineered features allow the model to better capture the underlying patterns and relationships in the data, leading to improved accuracy and predictive power.

Through this process, new features have been created to improve the quality of the dataset and enable the model to better understand the underlying patterns. These newly engineered features enhance the model's ability to learn effectively and contribute to its overall performance.The added features are outlined as follows:

1. **Length of Questions**: Measures the number of characters in each question, providing an indication of the question's complexity or verbosity.
2. **Number of Words in Both Questions**: Counts the total number of words in both questions, offering insights into their overall length and structure.
3. **Number of Common Words**: Calculates the number of words that are shared between the two questions, helping to assess their semantic similarity.
4. **Number of Total Words**: Represents the total number of words across both questions, reflecting the total content for comparison.
5. **Word Share (Number of Common Words / Total Words)**: Computes the ratio of common words to the total words across both questions, providing a normalized measure of similarity.

5 | Model Selection

A total of three distinct categories of models were employed to obtain the results. These include two similarity-based baseline models, a neural network model, and a pretrained large language model (LLM), all of which were implemented as part of the project.The models are as follows:

1. Similarity-based baseline models used : **TF-IDF** and **Doc2Vec**.
2. Neural Network based baseline model: **Long-Short Term Memory (LSTM)**.
3. Pretrained Transformer Model : **SIAMESE BERT**.

5.1 | Similarity Based Baseline Model - TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is a technique used to evaluate the importance of a word within a document relative to a corpus. It combines **Term Frequency**(how often a word appears) and **Inverse Document Frequency** (how common the word is across documents). This helps identify significant words, making it useful for text analysis, search, and document similarity tasks.

5.1.1 | Code Procedure

The following code procedure has been implemented for this model:

1. **Data Analysis** - Checking for null values, identifying duplicates, cleaning the text, and stemming words.
2. **Data Preparation** - Extracting Q1 and Q2 modules/lists from the input texts, filtering out invalid questions, and removing stop words.

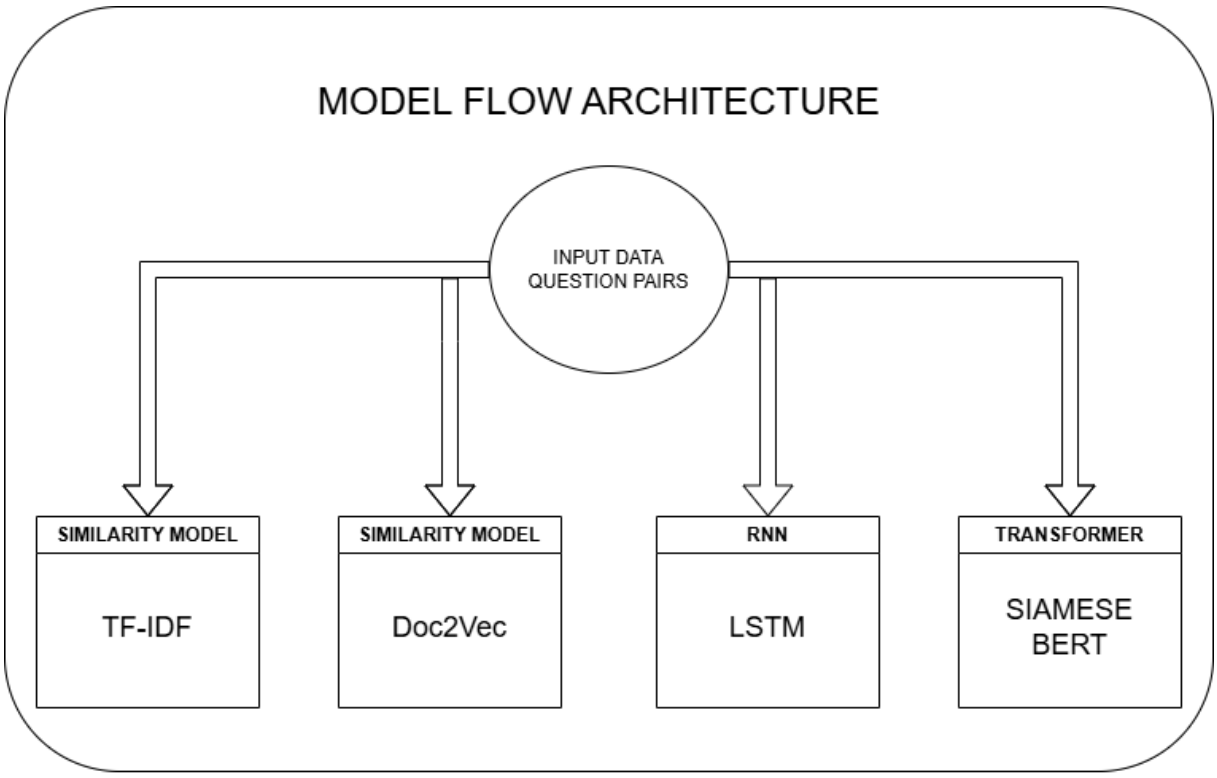

Figure 5.1: Model Flow Diagram

3. **Cosine Similarity Scores Calculation** - These scores are used by the model to classify whether the questions are similar or not.
4. **Number of Epochs** - set to 5, as it achieved the highest accuracy compared to other epochs.

5.1.2 | Limitations of TF-IDF Model

Although this model achieves notable accuracy in certain cases, its limitations hinder its broader applicability. Some of the limitations are as follows:

1. **Lack of Context**: TF-IDF treats words independently and doesn't capture their semantic meaning or context.
2. **Common Words**: It may still overemphasize domain-specific or frequently occurring words, even if they aren't informative.
3. **Sparsity**: The resulting vectors are sparse, leading to inefficiencies in storage and computation.
4. **No Word Order**: TF-IDF ignores word order, which can affect the meaning of phrases (e.g., "dog bites man" vs. "man bites dog").
5. **Scalability**: As the corpus grows, the TF-IDF matrix becomes large and computationally expensive to manage.

5.2 | Similarity Based Baseline Model - Doc2Vec

Doc2Vec is an extension of the Word2Vec model that generates fixed-length vector representations for entire documents, rather than just individual words. It works by learning a distributed representation of documents in a continuous vector space, where similar documents are mapped close to each other. The model processes data by first breaking down text into sentences and words, then training on these sequences using a neural network. It learns both word vectors

(as in Word2Vec) and a unique document vector for each input document. During training, Doc2Vec adjusts these vectors to predict context words within a document, ultimately capturing semantic meaning. This makes Doc2Vec particularly useful for tasks like document similarity, classification, and clustering.

5.2.1 | Code Procedure

The procedure for this model follows the same approach as the previously discussed similarity model (TF-IDF).

5.2.2 | Limitations of Doc2Vec Model

While Doc2Vec model generally outperforms TF-IDF in most cases, it has several key limitations, as outlined below:

1. **Data Dependency**: Requires large datasets for effective training; small datasets may yield poor embeddings.
2. **Preprocessing Sensitivity**: Performance is highly influenced by preprocessing steps, like tokenization and stop word removal.
3. **Computationally Intensive**: Training can be slow and resource-heavy with large corpora.
4. **Long-Range Dependencies**: Struggles to capture deep relationships in very long documents.
5. **Lack of Interpretability**: The document vectors are dense and hard to interpret.
6. **Task-Specific Performance**: May not perform well in domains where fine-grained semantic differences are crucial.

5.3 | Neural Network - Long Short Term Memory

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data. Unlike traditional RNNs, LSTMs address the vanishing gradient problem, allowing them to retain information over longer sequences. The model processes data by passing input sequences through its layers, where each LSTM cell has memory gates that regulate the flow of information. These gates—input, forget, and output—control how much of the previous information should be remembered or discarded. This makes LSTM particularly effective for tasks involving time-series data, language modeling, and sequence prediction, where the model needs to understand context from earlier time steps to make accurate predictions.

5.3.1 | Code Procedure

The LSTM code implementation follows these steps:

1. **Addition of New Features**
   - [a] Length of questions
   - [b] Number of words in both questions
   - [c] Number of common words
   - [d] Total number of words
   - [e] Word Share = (Common words / Total words)
2. **Data Visualization**
3. **Lemmatization and Model Run**

5.3.2 | Limitations of the LSTM Model

Long Short-Term Memory (LSTM) is a powerful recurrent neural network model that outperforms the similarity-based models mentioned earlier, offering higher accuracy and better effectiveness in processing input data. However, it does have some limitations, which are outlined below:

1. **Computationally Expensive**: Requires significant resources and time for training.
2. **Long Training Times**: Takes longer to train, especially on large datasets.
3. **Vanishing Gradient**: Struggles with very long sequences, despite improvements over RNNs.
4. **High Memory Usage**: Large memory footprint due to multiple gates and parameters.
5. **Overfitting**: Prone to overfitting, especially with small datasets.
6. **Complexity**: Difficult to tune and configure compared to simpler models.

5.4 | Pretrained Tranformer Model - SIAMESE BERT

A Siamese Neural Network (SNN) is a type of architecture that consists of two or more identical sub-networks. "Identical" refers to having the same configuration, parameters, and weights across both networks. These networks are designed to evaluate the similarity between two inputs by comparing their feature vectors. Inspired by the concept of Siamese twins—conjoined brothers Chang and Eng Bunker, born in Siam (now Thailand)—the architecture uses this twin setup to process two different inputs in parallel. This shared structure enables the model to learn from both inputs simultaneously, extracting features with the same settings.

Once the features are extracted, a distance layer (such as L1 or triplet loss) is applied to compute the similarity between the embeddings. This is followed by dense layers and a classification head for final prediction. For our problem, we implement two variations of the Siamese network: the original Siamese Network with an L1 distance layer and a Siamese Network with Triplet Loss. Both models are optimized to assess question pair similarity effectively.

5.4.1 | SIAMESE Architecture

Unlike the LSTM model, Siamese BERT employs a parallel computational approach, allowing it to process both input questions simultaneously. This parallelism enhances the model's efficiency, enabling faster computation and better handling of large datasets. As a result, Siamese BERT can achieve more accurate results with reduced computational overhead. The key features of the model architecture are as follows:

1. This architecture uses the BERT twin backbone and applies the L1 distance on the embeddings returned by the backbone.
2. The L1 distance features are then fed to a dense layer to capture the non-linearities.
3. The final layer is a sigmoid neuron which classifies whether the non-linear activated distance features indicate if the sentences are similar or dissimilar.

5.4.2 | Code Procedure

The implementation of the Siamese BERT model for Quora Question Similarity Detection follows these steps:

1. **Data Preparation and Exploration**
   - [a] **Data Cleaning**:
     - i. Remove null values.
     - ii. Perform basic cleaning (removal of special characters, extra spaces, and case normalization).

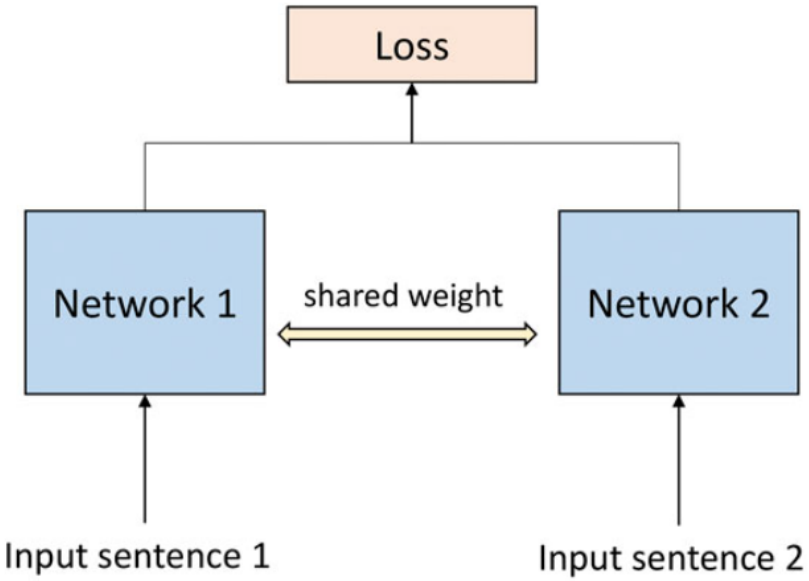

Figure 5.2: Siamese Architecture Diagram

[b] **Exploratory Data Analysis (EDA)**:

i. Visualize the distribution of duplicate vs. non-duplicate questions using pie charts.

ii. Analyze sentence lengths and identify an ideal sentence length for modeling.

2. **Text Preprocessing**

[a] **Text Cleaning Function**: Define a custom function to remove noise from the text while preserving semantic content.

[b] **Tokenization and Encoding**:

i. Use the BERT tokenizer (`AutoTokenizer`) to convert text into token IDs.

ii. Generate attention masks and padded encodings.

3. **Data Splitting**

[a] Sample 400,000 rows from the dataset.

[b] Split the data into training (80%) and validation (20%) sets.

4. **Siamese BERT Model Architecture**

[a] **Pre-trained BERT Backbone**:

i. Use the `bert-base-uncased` variant for transfer learning.

ii. Extract embeddings for question pairs using identical BERT encoder networks.

[b] **L1 Distance Layer**: Calculate the absolute difference between the embeddings of the two questions.

[c] **Dense Layers**:

i. Pass the L1 distance through dense layers with ReLU activation.

ii. Use a final dense layer with a sigmoid activation for binary classification.

5. **Training Configuration**

[a] **Batch Size and TPU Support**:

i. Configure the model for TPU training if available.

ii. Use a batch size of 32 if TPU is unavailable.

[b] **Callbacks**:

i. Implement early stopping to avoid overfitting.

ii. Use learning rate reduction on validation loss plateau.

6. **Model Training**

[a] Train the model for 5 epochs, tracking metrics like loss and accuracy on training and validation data.

7. **Evaluation and Visualization**

[a] **Learning Curves**:
i. Plot training and validation loss.
ii. Plot training and validation accuracy.

[b] **ROC-AUC Curve**:
i. Compute and visualize the ROC curve, reporting an AUC score of 95%.

[c] **Confusion Matrix**:
i. Generate a confusion matrix to analyze prediction performance.

[d] **Classification Report**:
i. Calculate precision, recall, F1-score, and overall accuracy (89%).

6 | Results and Error Analysis

6.1 | Similarity Based Baseline Models

Although similarity-based baseline models are expected to perform well, the output labels in the data have fewer features, which limits the models' ability to make accurate predictions, resulting in lower overall accuracy.

Table 6.1: "Accuracies of Similarity Models Across Data Points"

Data Points	EPOCHS	TF-IDF Accuracy	Doc2Vec Accuracy
50,000	5	64%	64%
200,000	5	64%	67%
400,000	5	65%	68%

6.2 | LSTM (RNN)

This neural network outperforms the previously discussed models (TF-IDF and Doc2Vec) due to its distinct computational approach, which relies on sequential computations. The output accuracies are presented in Table 6.2, while the training and validation losses are depicted in Figure 6.1.

Table 6.2: "Accuracy of LSTM Model Across Data Points"

Data Points	EPOCHS	LSTM Accuracy
200,000	5	73%
400,000	5	76%

6.3 | Pre-Trained Transformer Model - SIAMESE BERT

Siamese BERT achieved higher accuracy for Quora Question Pairs by leveraging its parallel computational approach, unlike the sequential LSTM model or the feature-based similarity models (TF-IDF, Doc2Vec). Using BERT's pre-trained transformer architecture, it captures deeper contextual relationships between questions. The model processes both questions simultaneously, compares their feature embeddings with an L1 distance layer, and efficiently identifies semantic similarities. This allows Siamese BERT to outperform both similarity-based models and LSTM in capturing complex sentence relationships.

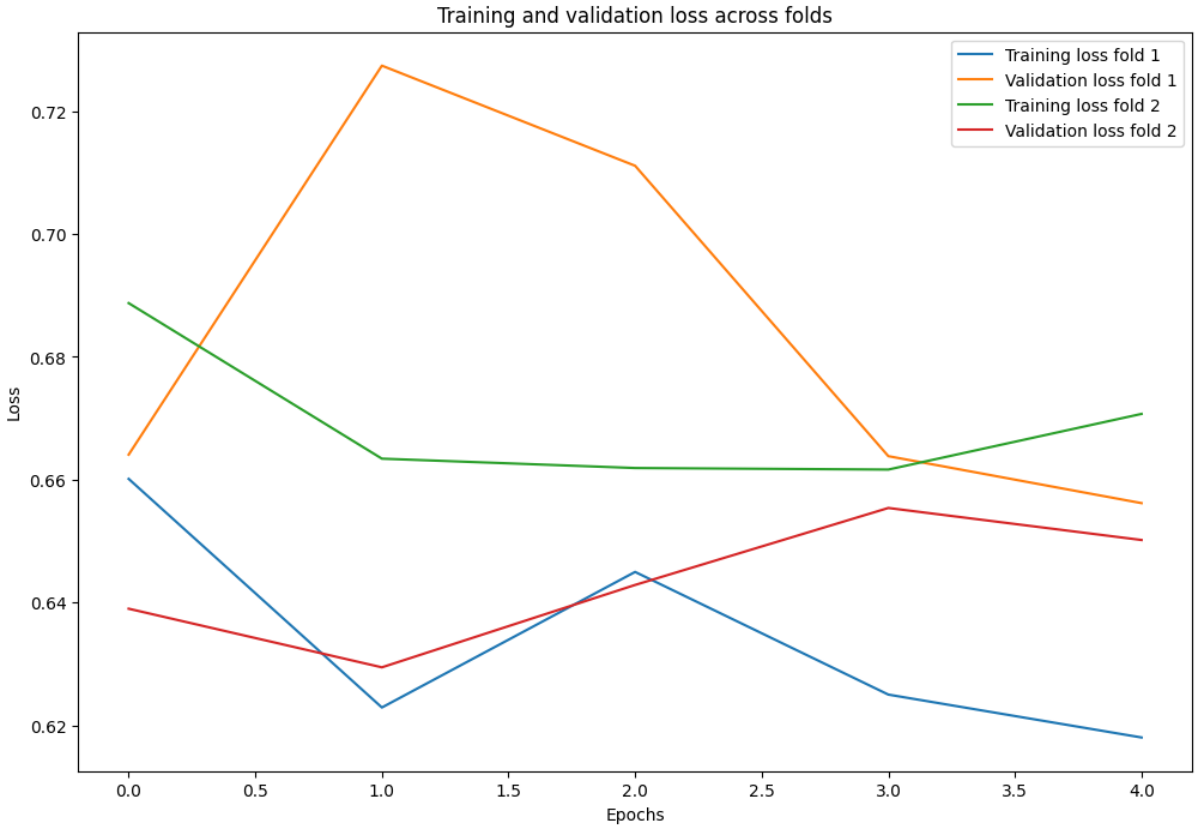

Figure 6.1: LSTM Model Losses

Table 6.3: "Accuracies of LSTM Model Across Data Points"

Data Points	EPOCHS	SIAMESE BERT Accuracy
200,000	3	86%
400,000	3	89%

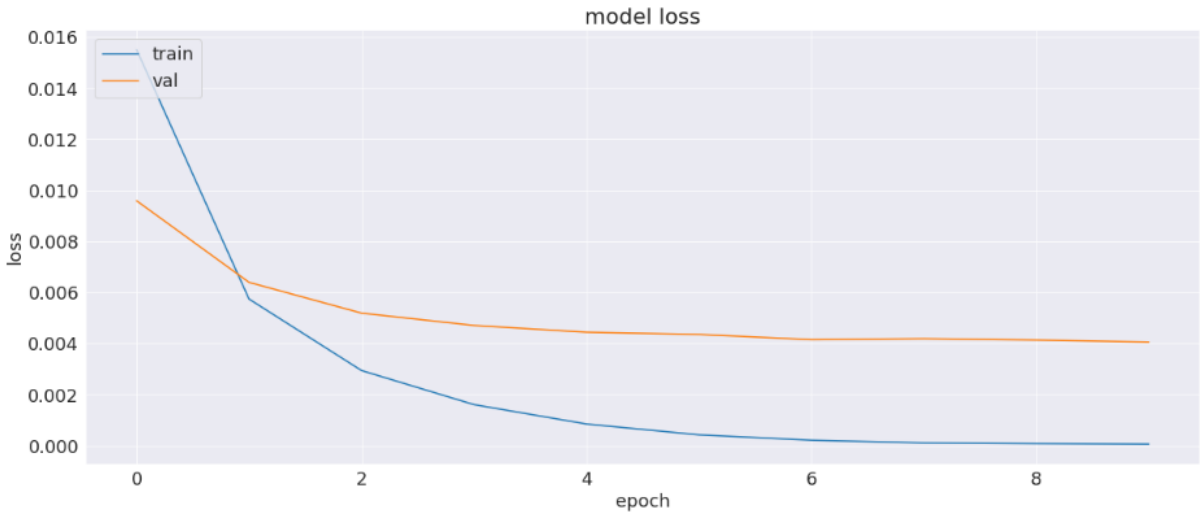

Figure 6.2: SIAMESE BERT Model Loss

6.4 | Inference

Based on the accuracy results as shown in table 6.4, the Siamese BERT model significantly outperforms the other models, achieving an impressive 89% accuracy. While TF-IDF, Doc2Vec, and LSTM models show decent performance (64%, 67%, and 76%, respectively), Siamese BERT demonstrates superior capability in identifying question similarities.

Table 6.4: "Accuracies of Various Models"

MODEL	ACCURACY
TF-IDF	65%
Doc2Vec	67%
LSTM	76%
SIAMESE BERT	89%

7 | Limitations and Challenges

While our original plan was to develop a comprehensive platform that would direct users to the top 5 most similar questions related to the query they submitted, we faced several limitations that restricted us to building only the current model. The main challenge was the lack of sufficient and high-quality data to train the system effectively. We envisioned a solution that could handle real-time user queries and provide relevant question suggestions based on advanced similarity models. However, due to constraints in the dataset—such as limited labeled data and difficulty in obtaining a broader range of question pairs—we were unable to proceed with the full-scale platform. As a result, we focused on building a model capable of classifying whether two questions are similar or not, a more feasible approach within the available resources.

8 | References

[1] Kaggle Inc. Quora question pairs dataset. https://www.kaggle.com/datasets/quora/question-pairs-dataset, 2024. Accessed: 2024-12-14.

[2] Quora Inc. Quora: A place to share knowledge and better understand the world. https://www.quora.com, 2024. Accessed: 2024-12-14.

[3] Huong T. Le, Dung T. Cao, Trung H. Bui, Long T. Luong, and Huy Q. Nguyen. Improve quora question pair dataset for question similarity task. *IEEE*, 2021. Accessed: 2024-12-14.

As discussed in [3], automatic detection of semantically equivalent questions plays a crucial role in a question answering system. The dataset used in this project is available on Kaggle [1]. Quora, the platform from which the dataset originates, is also central to this research [2].