Detecting Duplicate Queries on Quora: An NLP Approach to Similarity Analysis

EDS 6397 – Introduction to Natural Language Processing

Team 10: Venkata Kaushik Belusonti & Vishnu Sai Inakollu

1. Introduction

Quora is a widely known search platform in which millions of users across the globe post questions and seek answers to the questions. Many a times, users get disappointed as a result of duplicate questions posted on the platforms. This makes difficult for the users to get the answers they are seeking for as they might looking into the duplicate questions posted on Quora. Managing the Quora and avoiding the presence of duplicate questions in the Quora platform enhances the user experience and thus helps the in seeking the right information and as well posting their answers so that it is benefits the other users.

This project titles, Detecting the Duplicate Queries on Quora: An approach to Similarity Analysis aims in coming up with an approach or methodology to estimate the similarity of the questions. In this project, the model is developed in such a way that the model classifies the two questions into either Similar or Dissimilar. The user experience in Quora can be greatly enhanced by addressing the challenge of predicting the occurrence of the duplicate question in the Quora platform and enhancing the engagement in Quora community by removing the duplicate question.

2. Dataset selection

Selection of a dataset is a critical in evaluating the performance of model, be it a machine learning, deep learning or Natural language Processing for a specific task. The dataset shall be in the field of the specific taks so that the model can be well trained for a given task and will be able to perform well on unseen task. The following key considerations are to be taken into account for selecting the dataset.

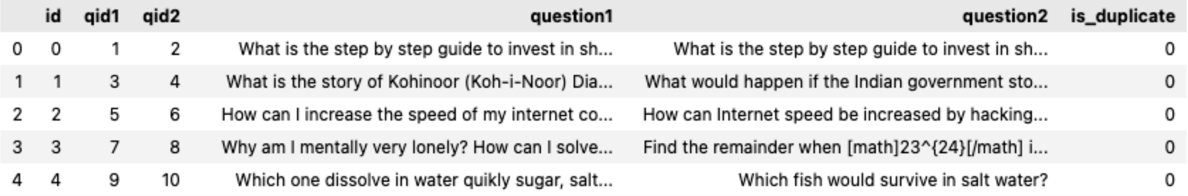
1) Relation of the dataset to the task: The dataset shall be related to Question pairs and will be perfect if the questions are straightaway taken from the Quora. The data shall be labelled and binary classified as similar and not similar.
2) Avoid incorrect labelling: The labelling quality shall be good. In case if the similarity of question pairs are incorrectly labelled, this introduces the noise into the dataset and the model training will be ineffective and. Look into the data and make sure to avoid the possibility of incorrect labelling. This can be achieved by selecting the dataset from the reliable source.
3) Size of the dataset: The model for performing the question similarity will be from deep learning, LSTM and BERT to name a few and will have higher number of parameters. Training of the model with higher number of parameters requires large dataset. The larger the dataset enables the model to learn complex patterns and the developed model ensures the reliable performance in predicting the similarity of the question pairs.
4) Compatibility: A careful selection of dataset based on the primary language of the intended selection will result in the effective model training. In this case, the objective is to estimate the similarity of question pairs in Quora platform. The questions in the Quora platform will be in English language. So, the dataset shall have the question pairs in English language to reliable model development. So, the developed model may not work effectively if the application is to perform question-pair similarity in anly language other than English.
5) Ethical consideration: The selected dataset shall always adhere to data privacy regulations and guidelines. The dataset shall not have any private information of the user and the government secret information and thus avoiding the unethical ways will ensure responsible use of the data.

With the above mentioned considerations, datasets are carefully evaluated. A close to 10 datasets are filtered in selecting the final dataset for predicting the question pair similarity. The selected dataset adhering all the above mentioned consideration is the Quora Question Pairs [1] taken from the Kaggle website. The same data has been in one of the competitions hosted by Kaggle in 2017.

The features of the dataset are as below.

1) Contains ~404,000 pairs of questions along with their associated IDS, including the information on whether each pair of questions is similar or not.
2) The first few rows of the dataset is as shown in the Table I. The columns in the dataset are explained below.
   a. *id*: Unique index number of the question pairs
   b. *qid1* and *qid2*: A unique identifier assigned to each question in the dataset.
   c. *question1* and *question2*: Questions of the question pairs in string format
   d. *is_duplicate*: A value of 1 indicates that the questions are similar, while a value of 0 signifies that the questions are not similar.

Table I: First Five Rows of the Dataset



	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

3) The distribution of the word length of the questions in the columns *question1* and *question2* is shown in the Fig. 1. The majority of the questions in columns *question1* and *question2* are having the word count with less than 40 words and following a uniform distribution.

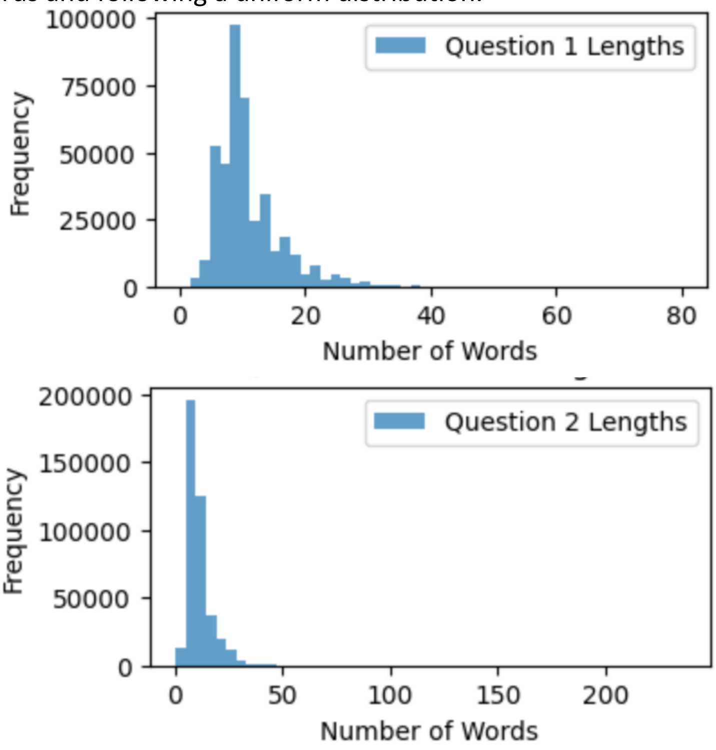


Fig. 1. Distribution of the word length of the questions in question1 and question2 columns

4) The distribution of the duplicate (similar) and non-duplicate (dissimilar) question pairs is shown in Fig. 2. The data is highly imbalanced.
5) The dataset contains two duplicate pairs and contains two question pairs with at least one question missing in the question pairs.
6) The dataset consists of many contractions. A well planned pre-processing to be performed in dealing with the contractions. The word *cannot* and *can't* are used in many question however both mean the same.
7) The dataset contains questions only in English language. This is good thing for training the model to perform well in predicting the similarity of question pairs in English language.
8) There are only two tuples with at least one missing value in the columns.

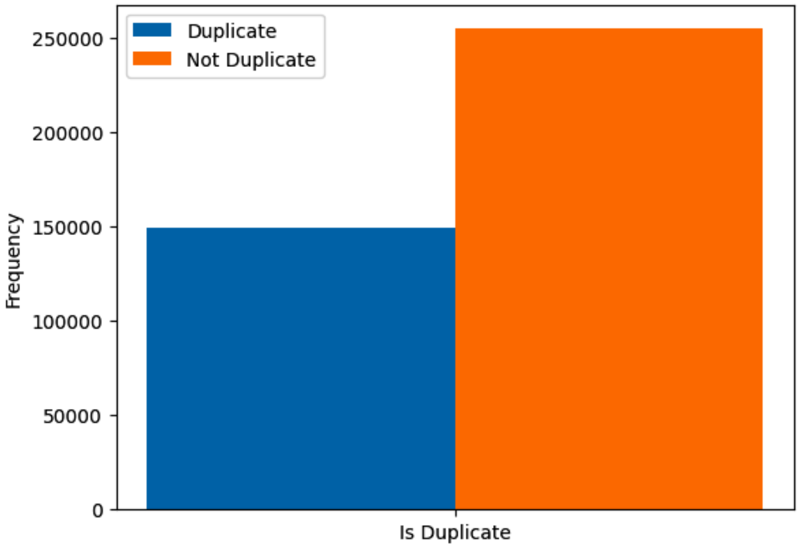


Fig. 2: Contribution of similar and dissimilar question pairs in the dataset

3. Methodology

The objective of this project is to utilize the learnings from EDS6397 – Introduction to Natural language Processing course in predicting the similarity of question pairs in Quora platform. The models from Natural Language Processing (NLP) are selected and are trained to perform the mentioned task. Three distinct models are finalized to achieve the task. The project discusses the selection of the dataset, pre-processing of the data to ensure the data is clean and ready for the model training. Then the data after pre-processing is used to train all the three models independently.

The model shall take the questions from the question pair as the input and classify the question pair into either of the binary classes, *Similar* and *Dissimilar*. As the task is classification, the performance metric selected for evaluating the models are confusion matrix metrics, Accuracy, Precision, Recall, and F1-score. These performance metrics gives the model view of estimating the true positives, true negatives, false positives, and false negatives. These metrics are widely known for their capability in estimating the quality of trained model and can be used as judgement for releasing the model for deployment in real-time applications. The better model of three will be selected of three by evaluating the above mentioned metrics.

4. Data Pre-processing

The below data pre-processing steps are performed before feeding the data to the model.

1) Keep only the needed columns: The model shall requires the questions of the question pair in text format and similarity of the questions be it a categorical or numerical. Therefore, barring the mentioned three columns, all other columns are removed from the dataset. The final set of columns in the dataset are *question1*, *question2*, and *is_duplicate.*
2) Remove duplicate tuples: Two duplicate tuple sets present in the dataset are identified and the duplicate tuples are removed.
3) Remove tuples with at least one missing cells: There are only two tuples with at least one missing cell in the enormous dataset. Removal of just two tuples will have insignificance influence on the model training. So, the two such tuples are removed.
4) Maintaining word casing: The casing of the word can influence the semantics of a given sentence. For example, the word “Apple” can refer to a company or fruit wherein the word, “apple” refers to only fruit. This distinction is important for estimating the similarity of questions in the question pair.
5) Remove HTTP URL: The website links will not be useful in developing the model to predict the similarity of the question pair. All URLs in the dataset are removed from the dataset.
6) Punctuation handling: Punctuations in a sentence will have semantic information and improper usage of punctuation will lead to the different meaning of the sentence. For example, “Let’s eat, grandma” and “Let’s eat grandma” can completely the alter the semantics in the sentence. Hence, punctuations are not removed from the dataset.

5. Models

Three models are selected for the estimating the similarity of the questions in the question pair. Each model will be briefly discussed highlighting the architecture, advantages and disadvantages in relevance to the task. The final aim is to come up with the most appropriate model of three in predicting the question pair similarity.

5.1. Baseline Model

The architecture of the baseline model to predict the similarity of the question pairs for the given question pair dataset is shown in Fig. 3.

The questions of the question pair are tokenized using Spacy Small English Language model. The word embedding of the tokens in the given question are estimated using static word embedding model. The selected static word embedding model is Glove (Global Vectors for Word Representation) 6billion 300 dimensional word embedding model [2] is selected for its co-occurrence in larger corpus of data. The word embeddings of all tokens in a question are predicted using GloVe model and these embedding are averaged for a given sentence. Similarly the average of the word embedding of the other question in the question pair is estimated. Upon estimating the average of the word embedding of two questions in the question pair, the cosine similarity, value ranges from -1 to +1 is computed between the average word of the word embeddings of the two questions in the question pair. Cosine similarity of 1 indicates the questions are exact match and the value very closer to -1 indicates exact opposite.

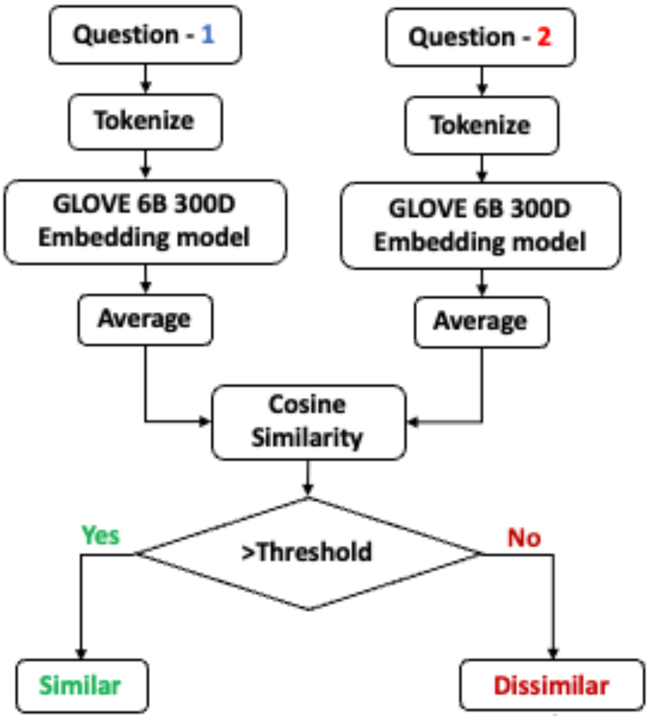


Fig. 3. Flow chart of baseline model used to predict the similarity of question pairs

The cosine similarity score can be acted as a threshold for deciding the decision boundary in deciding the similarity of the questions in question pair. A threshold value needs to be decided in classifying the question pair as *Similar* and *Dissimilar*. The question pair can be classified as *Similar* if the estimated Cosine similarity score is greater than or equal to threshold value, and *Dissimilar* if otherwise. The threshold value is tuned using the question pairs from the training data to maximise the accuracy, precision, recall and f1-score performance metrics. The selection of appropriate threshold value maximises the trustability of model's prediction in similarity of question pairs.

The reason for the selection of GloVe static word embedding model from the available static word embedding models is the strike balance between computation efficiency and ability to capture semantic information.

5.2. GloVe + Long Short-Term Memory (LSTM) with Siamese Neural Network

The Glove [2] static word embedding model when combined with Long Short-Term Memory (LSTM) [3] with Siamese Neural Network [4] is known be one of the effective methods in estimating the similarity of the questions in the question pair. This approach leverages the LSTM model known for its sequential pattern related model and utilizes the comparison capability of the Siamese Neural Network to capture the semantic similarity.

A Siamese neural network consists of two identical networks with shared parameters. The symmetry feature of the Siamese neural network ensures that the two inputs are processed in an identical manner, enabling the model to do a perfect work of comparing the inputs in terms of semantic behaviour.

The model architecture is shown in Fig. 4. The questions of the question pair are tokenized using Spacy Small English Language model. The word embedding of the tokens in the given question are estimated using static word embedding model. LSTM does not have inbuilt word embeddings. The selected static word embedding model is Glove (Global Vectors for Word Representation) 6billion 300 dimensional word embedding model [2] is selected for its co-occurrence in larger corpus of data. The word embeddings of all tokens in a question are predicted using GloVe model and these embedding and the embeddings are passed through the LSTM model of the parallel networks in Siamese Neural Network. Note that the model parameters of LSTM will be same in the Siamese Neural Network. The output of the LSTM models from the Siamese Neural network are used to compute the L2 (Manhattan) distance between the questions. The computed L2 distance is passed through a single forward layer followed by a sigmoid activation function to classify the questions as *Similar* and *Dissimilar*. Note that all the LSTM model parameter are frozen during the training and only the weights and biases of the forward layer is used for model training.

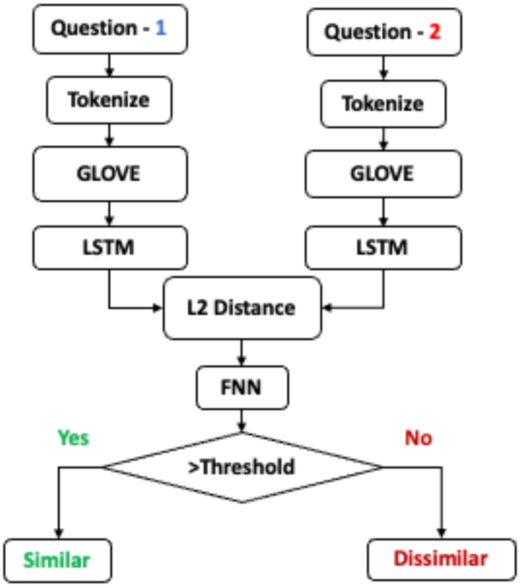


Fig. 4. Flow chart of Long Short-Term Memory (LSTM) with Siamese architecture used to predict the similarity of question pairs

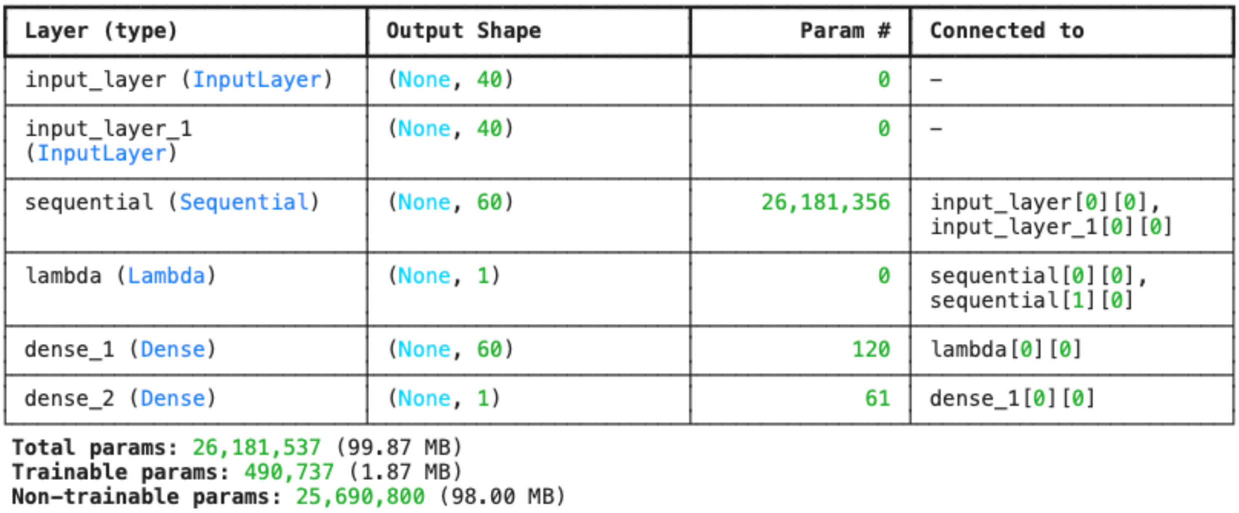
The snapshot of the model summary is shown in Table II. The input layers named *input_layer* and *input_layer1* represents two inputs of the Siamese Neural Network. Each input takes 40 tokens in each time step. As mentioned earlier, both questions are processed independently in two identical model layers, as the model employs Siamese architecture.

The sequential layer mentioned in Table II represents the layer with LSTM model in both parallel networks. The 26,181,356 parameters indicate the complex nature of the LSTM model and involves many LSTM units or layers to encode the sequential information effectively. All of 26,181,356 LSTM parameters are frozen during model training.

After the LSTM layer, the *lambda* layer of the shared sequential layer computes the L2 distance between the outputs of identical parallel LSTM models. The output of the *lambda* layer is connected fully connected network of two layers. The first layer of the fully connected network consists of 60 neurons with 120 trainable parameters (60 weights and 60 biases). The second layer contains only one neuron with sigmoid activation function to classify the question pair as *Similar* or *Dissimilar*. The second layer has 61 trainable parameters (60 weights and one bias).

Importantly, all LSTM parameters are frozen (non-trainable), while the weights and biases of the fully connected layers are trainable.

Table II: Model summary of LSTM with Siamese Neural Network



Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 40)	0	-
input_layer_1 (InputLayer)	(None, 40)	0	-
sequential (Sequential)	(None, 60)	26,181,356	input_layer[0][0], input_layer_1[0][0]
lambda (Lambda)	(None, 1)	0	sequential[0][0], sequential[1][0]
dense_1 (Dense)	(None, 60)	120	lambda[0][0]
dense_2 (Dense)	(None, 1)	61	dense_1[0][0]

```
Total params: 26,181,537 (99.87 MB)
Trainable params: 490,737 (1.87 MB)
Non-trainable params: 25,690,800 (98.00 MB)
```

5.3. Bidirectional Encoder Representations from Transformers (BERT) with Siamese Neural Network

Siamese Neural Network leveraging the Bidirectional Encoder Representations from Transformers (BERT) is known to be a proven architecture in the predicting the similarity of the two texts in real-time applications and the architecture is illustrated in Fig. 5. BERT base uncased model [5] downloaded from Hugging Face is used. A Siamese neural network consists of two identical networks with shared parameters.

The symmetry feature of the Siamese neural network ensures that the two inputs are processed in an identical manner, enabling the model to do a perfect work of comparing the inputs in terms of semantic behaviour.

The questions of the question pair are fed into the model of identical models in the Siamese neural network. Each question is tokenized using the inbuilt BERT tokenizer and passed through the identical pre-trained BERT models that share same weights. The BERT base uncased model generated the contextual dynamic embeddings for each token in the question. The dynamic embeddings obtained from the BERT model are pooled using average pooling to produce a fixed length vector representation for each question.

After the average pooling layer, the output of the average pooling layer is connected to five fully connected layers with 512 neurons, 256 neurons, 256 neurons, 64 neurons and one neuron for five layers respectively. These layers progressively transform the feature space, applying Batch Normalization and Dropout at each step to ensure and prevent overfitting.

All the parameters of BERT are frozen (no-trainable) and the weights and biases of five fully connected layers are trainable.

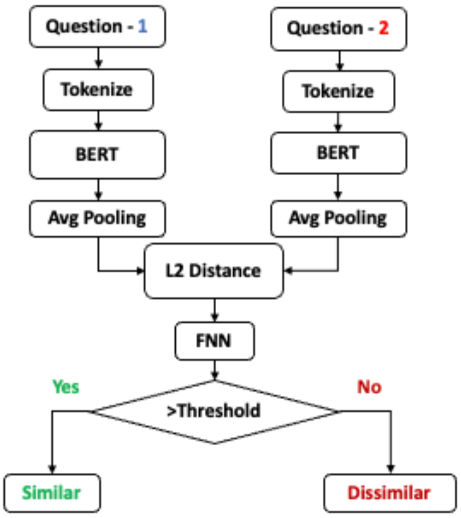
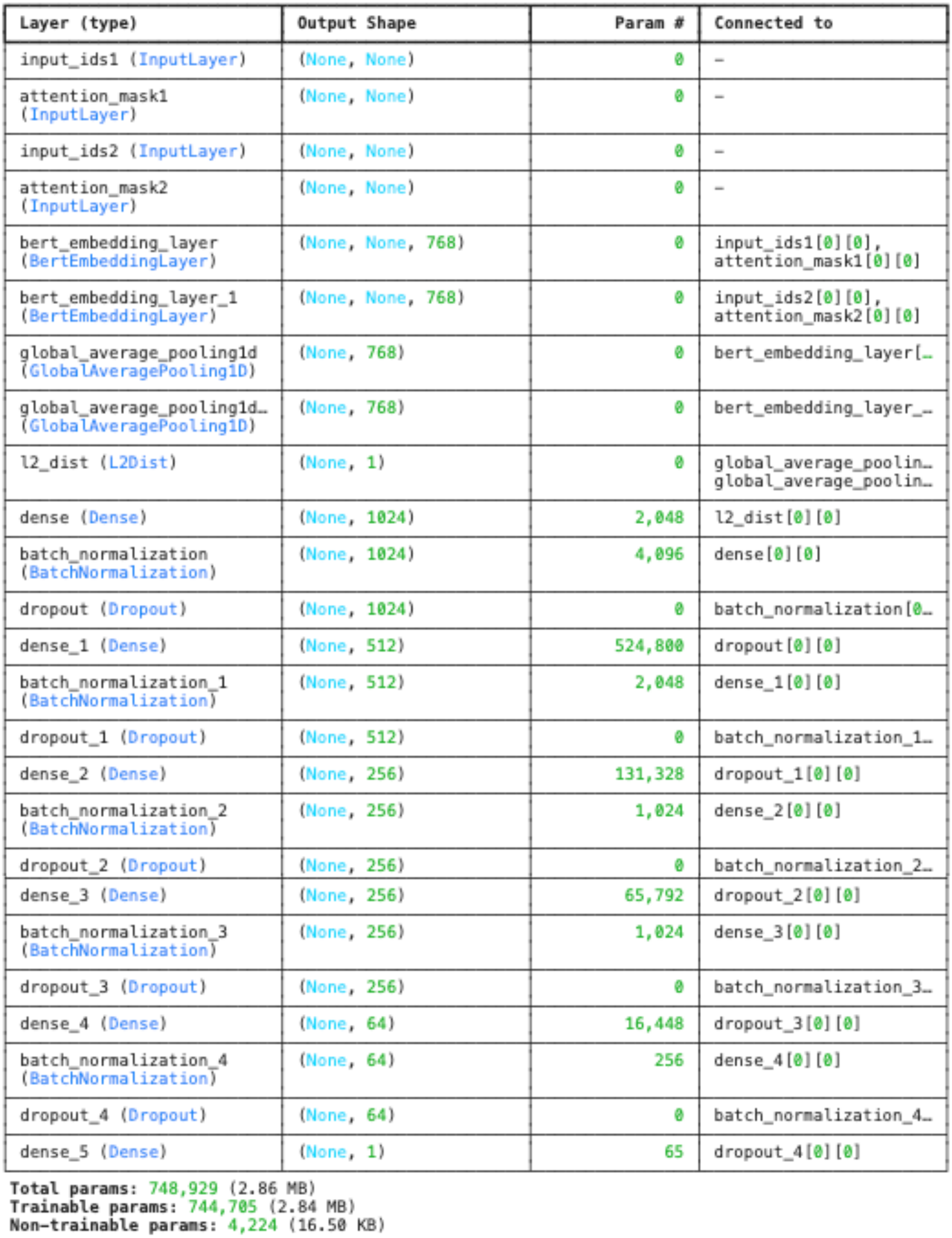


Fig. 5. Flow chart of BERT with Siamese Neural Network used to predict the similarity of question pairs

Table III: Model summary of BERT with Siamese Neural Network



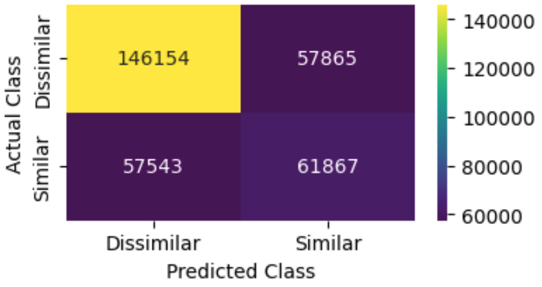
Layer (type)	Output Shape	Param #	Connected to
input_ids1 (InputLayer)	(None, None)	0	-
attention_mask1 (InputLayer)	(None, None)	0	-
input_ids2 (InputLayer)	(None, None)	0	-
attention_mask2 (InputLayer)	(None, None)	0	-
bert_embedding_layer (BertEmbeddingLayer)	(None, None, 768)	0	input_ids1[0][0], attention_mask1[0][0]
bert_embedding_layer_1 (BertEmbeddingLayer)	(None, None, 768)	0	input_ids2[0][0], attention_mask2[0][0]
global_average_pooling1d (GlobalAveragePooling1D)	(None, 768)	0	bert_embedding_layer[…
global_average_pooling1d… (GlobalAveragePooling1D)	(None, 768)	0	bert_embedding_layer_…
l2_dist (L2Dist)	(None, 1)	0	global_average_poolin… global_average_poolin…
dense (Dense)	(None, 1024)	2,048	l2_dist[0][0]
batch_normalization (BatchNormalization)	(None, 1024)	4,096	dense[0][0]
dropout (Dropout)	(None, 1024)	0	batch_normalization[0…
dense_1 (Dense)	(None, 512)	524,800	dropout[0][0]
batch_normalization_1 (BatchNormalization)	(None, 512)	2,048	dense_1[0][0]
dropout_1 (Dropout)	(None, 512)	0	batch_normalization_1…
dense_2 (Dense)	(None, 256)	131,328	dropout_1[0][0]
batch_normalization_2 (BatchNormalization)	(None, 256)	1,024	dense_2[0][0]
dropout_2 (Dropout)	(None, 256)	0	batch_normalization_2…
dense_3 (Dense)	(None, 256)	65,792	dropout_2[0][0]
batch_normalization_3 (BatchNormalization)	(None, 256)	1,024	dense_3[0][0]
dropout_3 (Dropout)	(None, 256)	0	batch_normalization_3…
dense_4 (Dense)	(None, 64)	16,448	dropout_3[0][0]
batch_normalization_4 (BatchNormalization)	(None, 64)	256	dense_4[0][0]
dropout_4 (Dropout)	(None, 64)	0	batch_normalization_4…
dense_5 (Dense)	(None, 1)	65	dropout_4[0][0]

Total params: 748,929 (2.86 MB)
Trainable params: 744,705 (2.84 MB)
Non-trainable params: 4,224 (16.50 KB)

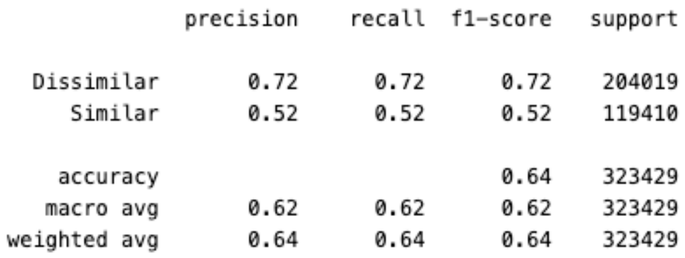
6. Results

This section discusses the performance of three models in predicting the similarity of question pairs in performance metrics, confusion matrix metrics, accuracy, precision, recall and f1-score.

In baseline model, the data is split into training and testing with 80:20. The threshold of the baseline model is tuned using the training data. The confusion matrix and associated performance metrics for the baseline model for the training data is shown in Fig. 6.



a. Confusion matrix



```
              precision    recall  f1-score   support

  Dissimilar       0.72      0.72      0.72    204019
     Similar       0.52      0.52      0.52    119410

    accuracy                           0.64    323429
   macro avg       0.62      0.62      0.62    323429
weighted avg       0.64      0.64      0.64    323429
```

b. Classification report

Fig. 6. Baseline Model: Confusion matrix and classification report on the training data

The GloVe plus LSTM with Siamese Neural Network is trained for 15 epochs. The convergence of model loss with respect to the number of epochs is illustrated in Fig. 7. The performance metrics, confusion matric, precision, recall, f1-score and accuracy for predicting the Similar and Dissimilar classes is shown in Fig. 8.

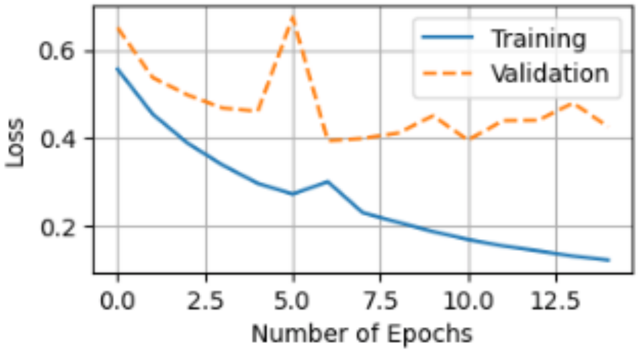
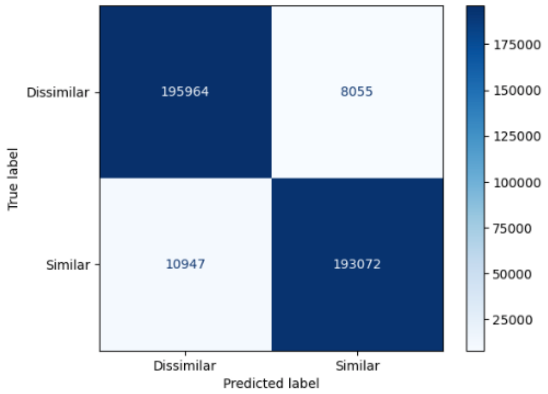
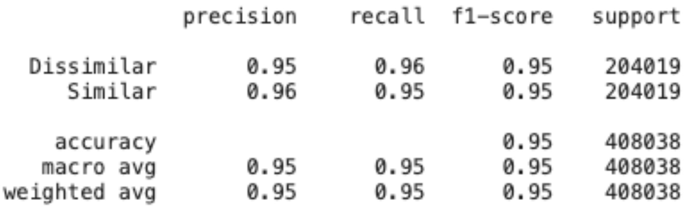


Fig. 7. LSTM with Siamese Network: Model loss with respect to epoch





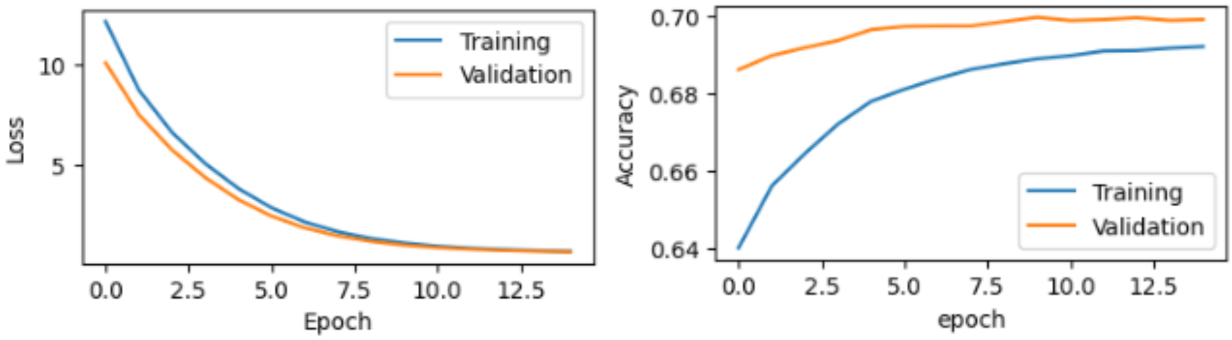
```
              precision    recall  f1-score   support

  Dissimilar       0.95      0.96      0.95    204019
     Similar       0.96      0.95      0.95    204019

    accuracy                           0.95    408038
   macro avg       0.95      0.95      0.95    408038
weighted avg       0.95      0.95      0.95    408038
```

a. Confusion matrix

b. Classification report

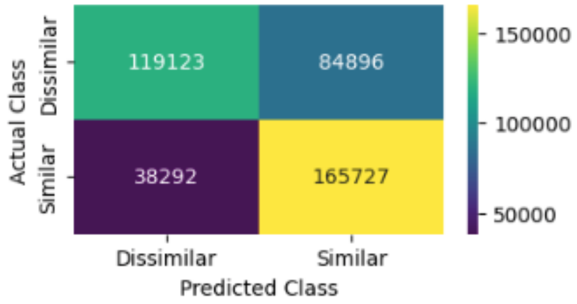
Fig. 8. LSTM with Siamese Network: Confusion matrix and classification report on the training data

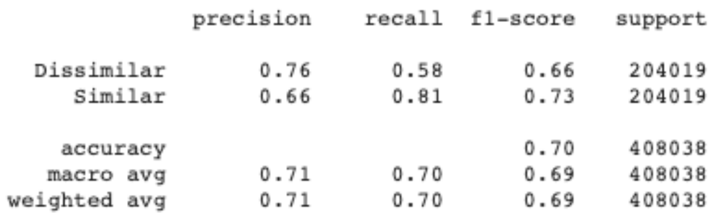
BERT with Siamese Neural Network is also trained for 15 epochs. The model loss and accuracy with respect to epochs on the training data is shown in Fig. 9.



a. Model loss b. Model accuracy

Fig. 9. BERT with Siamese Network: Model loss and accuracy with respect to epoch

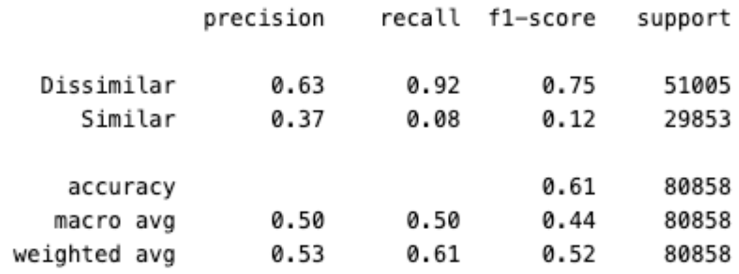




```
              precision    recall  f1-score   support

   Dissimilar       0.76      0.58      0.66    204019
      Similar       0.66      0.81      0.73    204019

     accuracy                           0.70    408038
    macro avg       0.71      0.70      0.69    408038
 weighted avg       0.71      0.70      0.69    408038
```

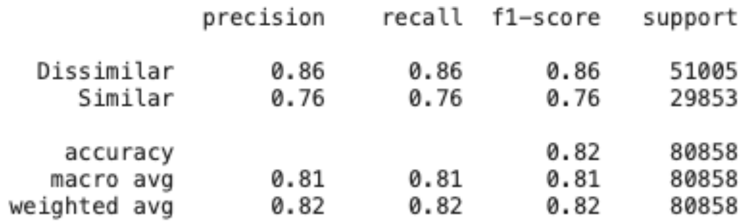
Fig. 10. BERT with Siamese Network: Confusion matrix and classification report on the training data

The performance of the baseline, LSTM with Siamese Network and BERT with Siamese Network on unseen data is evaluated using classification report is shown in Fig. 11.



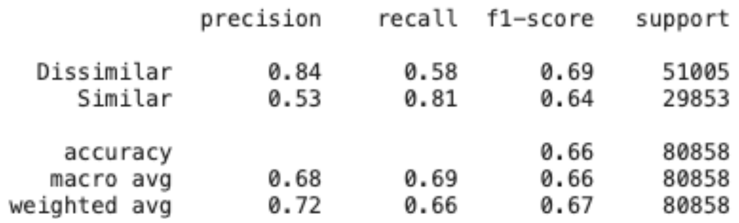
```
              precision    recall  f1-score   support

   Dissimilar       0.63      0.92      0.75     51005
      Similar       0.37      0.08      0.12     29853

     accuracy                           0.61     80858
    macro avg       0.50      0.50      0.44     80858
 weighted avg       0.53      0.61      0.52     80858
```

a. Baseline Model



```
              precision    recall  f1-score   support

   Dissimilar       0.86      0.86      0.86     51005
      Similar       0.76      0.76      0.76     29853

     accuracy                           0.82     80858
    macro avg       0.81      0.81      0.81     80858
 weighted avg       0.82      0.82      0.82     80858
```

b. LSTM with Siamese Network



```
              precision    recall  f1-score   support

   Dissimilar       0.84      0.58      0.69     51005
      Similar       0.53      0.81      0.64     29853

     accuracy                           0.66     80858
    macro avg       0.68      0.69      0.66     80858
 weighted avg       0.72      0.66      0.67     80858
```

c. BERT with Siamese Network

Fig. 11. Performance of baseline, LSTM with Siamese Network & BERT with Siamese Network on test (unseen) data

7. Discussion

The baseline model achieved 64% of training accuracy which is mediocre on both *Dissimilar* and *Similar* classes. Even though the performance is mediocre the F-1 score is 0.62 over all and for the similar class 0.52 is low which means the model is struggling to identify the Similar class, this discrepancy might be due to high imbalance of the training data, which contains close to 50% more samples of Dissimilar samples. Adding to it the threshold value of the base model which classifies the question pairs is also heavily influenced by the dominant Dissimilar class and this leads to the bias in the training model this is why the model struggles to classify. The model is performing very poorly on the similar class, the F-1 score for the similar class is close to 0.1 which means that the performance of the model is not considerable to use.

The LSTM with Siamese Neural Network model achieved significantly higher performance compared to the base model, with and accuracy of 95% on train data and the F-1 score(0.95), precision and recall across both *Dissimilar* and *Similar* classes are high which means that the model is learning on the training data quiet well. This is because LSTM model is very good at sequential contextual learning compared to the base model and this is important for our use case. On testing data, the F-1 score has dropped slightly, we have less samples for our similar class and we decided to not balance/ to have the same number of samples for both the classes this is because we never know how the testing data is going to be in the real time data. But the performance of the LSTM model is still good when it comes to testing data this says that the model has good capabilities of generalizing.

The BERT model with Siamese achieved reasonable F-1 score for both the classes but the similar class has a little higher score than the dissimilar class. On testing data, BERT model performed better than the base model when it comes to classifying both the classes, the F-1 scores for Dissimilar and Similar classes are reasonable but not better than LSTM model. The base model cannot learn the contextual relationship between the text whereas the LSTM and the BERT model can learn the contextual relation, the BERT model we used is not tuned on our data and the weights we used are pretrained weights so the model is not performing at its fullest for our use case but the LSTM model was trained on our data so the model was able to perform better for our case. But the BERT model is less biased towards the classes compared to the LSTM model.

8. Conclusion

Similarity prediction is a frequently seen issue in the real-time application, be in identifying the duplicate questions or search optimization. Baseline model, GloVe plus LSTM with Siamese Neural Network and BERT with Siamese Neural Network models are trained on the Question Pairs dataset taken from one of the Kaggle competitions. GloVe plus LSTM with Siamese Neural Network demonstrates the best performance among three models across both training and testing datasets, achieving a good balance of precision, recall, and F1-scores, along with strong generalization to unseen data. Despite its sophisticated architecture and is known for contextual classification, BERT with Siamese Neural Network performance is hampered by class imbalance or potential misalignment between its pre-training objectives and the task. As expected, the baseline model falls short of practical application due to poor contextual generalization and weak performance on the low sized class, *Similar* class. While LSTM with Siamese Neural Network in this report provides an excellent trade-off, transformer-based models like BERT lead in tasks requiring fine-grained semantic understanding, albeit with higher resource demands.

9. References

[1] Quora Question Pairs Dataset, Kaggle. 2017. [Online]. Available: https://www.kaggle.com/competitions/quora-question-pairs *[Accessed: Dec. 2, 2024].*

[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2024. GloVe: Global Vectors for Word Representation. [Online]. Available: https://nlp.stanford.edu/projects/glove/ *[Accessed: Dec. 2, 2024].*

[3] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd ed. Draft, p. 120, Fig. 8.13, 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/ *[Accessed: Dec. 2, 2024].*

[4]"Siamese Neural Network" Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Siamese_network. *[Accessed: Dec. 2, 2024].*

[5] BERT Base Uncased model from Hugging Face. [Online]. Available: https://huggingface.co/google-bert/bert-base-uncased *[Accessed: Dec. 2, 2024].*