

PROJECT REPORT
ON
SIMPLIFIER - INTELLIGENT QUESTION PAIR ANALYSIS
USING DEEP LEARNING

Submitted by

ROVAL BENNY (SJC17CS085)

ROYAL BENNY (SJC17CS086)

RICHU JOY (SJC17CS077)

SEBIN BYJU (SJC17CS094)

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the degree

of

Bachelor of Technology

in

Computer Science and Engineering



Department of Computer Science and Engineering
St. Joseph's College of Engineering and Technology, Palai

June :: 2021

Declaration

We undersigned hereby declare that the project report on ” **Simplifier - Intelligent Question Pair Analysis using Deep Learning** ”, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by us under supervision of Dr. Gemini George. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University

Name and Signature of Student

Roval Benny

Royal Benny

Richu Joy

Sebin Byju

Place: Choondachery

Date: 15-06-2021

ST. JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the report entitled "**Simplifier - INTELLIGENT QUESTION PAIR ANALYSIS USING DEEP LEARNING**" submitted by **ROVAL BENNY (SJC17CS085) ROYAL BENNY (SJC17CS086) RICHU JOY (SJC17CS077) SEBIN BYJU (SJC17CS094)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by her under my guidance and supervision.

Project Guide

Dr.Gemini George
Assistant Professor
Dept. of CSE

Project Coordinator

Ms. Angitha George
Assistant Professor
Dept. of CSE

.

Place: Choondachery

Date: 15-06-2021

.

Head of Department

Dr. Joby PP
Professor & Head
Dept. of CSE

Abstract

Today, the world is driven by data and many new technologies and product are in forefront to provide right information to users instantly. Everyday, we search for thousands of queries related to each domain and that too, in entirely different way. Based on regions and other factors, the pattern of question changes (semantics) but may have exactly same meaning. This brings a challenge to store answer for every such patterns within the server in order to provide the right answer. So our idea is to analyse two questions and extract certain parameters to determine the similarity of those questions. The deep learning technology is used in the project to find the similarity. Instead of checking for statement similarity, it analyse the sentence intelligently to identify accurately, the meaning of the questions and to output the similarity. Thus by identifying the similarity, it is very useful in many domains that enables them to store only a single copy of the answer for multiple questions having the same meaning. It can also be employed in chat bots and automated answer evaluation software where faculty can check the similarity of answer written by students to the exact answer, thus given marks accordingly. There emerges more domains where this project has significant role in eliminating the redundancy and making daily life more progressive.

Contents

Declaration	iii
Abstract	v
1 Introduction	1
1.1 Background	1
1.2 Objective and Scope	2
2 Literature Review	4
2.0.1 Quora Question Pairs Similarity Using Logistic Regression And Support Vector Machine	4
2.0.2 Exploring Deep Learning In Semantic Question Matching	5
2.0.3 A Model Based On Dual-Layer Attention Mechanism For Semantic Matching	5
2.0.4 Research On Question Answering System Based On Bi-LSTM And Self-Attention Mechanism	6
3 System Study	7
3.1 Modules	7
3.1.1 Query Input Module	7
3.1.2 Pre-processing Module	8
3.1.3 Feature Engineering Module	8
3.1.4 Training Module	8
3.1.5 Testing Module	9
3.1.6 Output Generation Module	9

Contents	vii
4 System Design	10
4.1 Use Case Diagram	10
4.2 Sequence Diagram	10
4.3 Architectural Diagram	12
4.3.1 Training Architecture	12
4.3.2 Testing Architecture	13
5 Project Design and Implementation	15
5.1 Design	15
5.1.1 Data Collection	15
5.1.2 Data Preprocessing	16
5.1.3 Feature Engineering	16
5.1.4 Model Creation	16
5.1.5 Output Generation	17
5.2 Implementation	17
5.2.1 Data Preprocessing	17
5.2.2 Feature Engineering	18
5.2.3 Model Creation	21
5.2.4 Output Generation	22
6 Experimental Results	24
6.1 Training Data	24
6.2 Training Result	25
Conclusion	29
References	30

List of Figures

1.1	Conventional Question Answering Methodology	2
4.1	Use Case Diagram	11
4.2	Sequence diagram	12
4.3	Training Architectural diagram	13
4.4	Testing Architectural diagram	14
5.1	Training Data Graph	18
5.2	Text Classification Bar Graph	19
5.3	NLP Features Bar Graph	21
5.4	Model Generation	22
6.1	Training Data	24
6.2	Accuracy Graph	25
6.3	Log-Loss Graph	26
6.4	Confusion Matrix	26
6.5	Evaluation Result	27
6.6	Home Page	27
6.7	When Questions Entered	28
6.8	Similarity Between Questions	28

Chapter 1

Introduction

Information is wealth. In a fast paced world, getting information which is accurate and within the context is necessary. And when we come to realize the speed and volume of data generated every single seconds, we should realize the need of a system that eliminates the redundancy in processing the information. More specifically, the redundancy of storing multiple copies of answer for the question having exactly the same meaning and in areas where we need to check the similarity of answers or statement. Many research is going on this domain which are trying to eliminate the issue

1.1 Background

The time when search engines were emerging to the market, it functioned by the text similarity of queries. It has no capability to understand the question meaning to retrieve answer. The number of people depending on internet for information is growing every day. These search engines takes user input in natural language form and the system will output a number of streamlined result or list of possible answers. The conventional methods seem to be inadequate in this current environment.

Thus there emerged a need for employing the advanced technology to understand the meaning of each input query to give the proper and updated answer to the user. For that we need a system that can find the similarity between two questions or statements. Moreover, every such server will be storing copies of answer multiple time for each questions

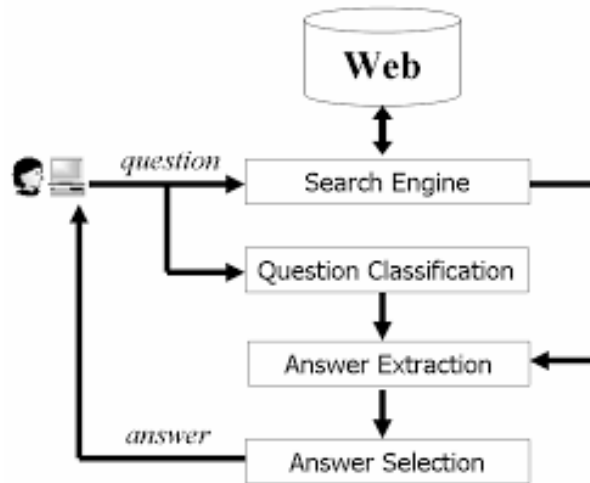


Figure 1.1: Conventional Question Answering Methodology

that are semantically similar but have different structure. This is causing additional overhead of memory and redundancy within the whole system. Also there are many domains where the exact meaning of the question or a statement is needed which can simplify human efforts. We will explain more such areas where our project can create a milestone in the scope of project.

1.2 Objective and Scope

Fig 1.1 above is the diagrammatic representation of how a conventional question answering works. From that it is clear that it simply classify questions and output the answers that probably relates to the question. In today's world, we can imagine that it is not worth. But at present more and more advancement has taken place in this direction and is still counting

Deep learning is one of the recent technology that came into play as Artificial Intelligence starts leading the technology. Our project address the issue with the help of deep learning. We should obtain two questions or statements from the user of which similarity will be calculated. This is the user input for the system. The next step is to do feature engineering on the two input queries and the result will be send for pre-processing. Pre-processing is done on the user input in order to remove the abnormalities and redundancy

to generate a common standard statements. Followed by pre-processing, the result is converted to word2vec. This vector assigns various value to each word.

Initially, we had a dataset that is officially Quora Question Pair dataset. We do feature engineering on the dataset to retrieve the features as many as possible. This will be saved for further analysis. Then we will normalize the feature engineering done on dataset and that of input questions will be normalized and is input to the model along with the word2vec vector. Thus the model has three inputs, two from the pre-processed user input and one from the normalized result of two feature engineering output. The model will do further processing and the final result, that is, the similarity of two questions or statements will be displayed. The user interface is done with Flask micro web framework.

The project has a great scope in various domains. Any area that requires natural language understanding, the project is super useful. As mentioned in the introduction, the major scope of this project comes in search engine industry where redundancy of storing multiple copies of the answer for questions that has the same meaning but different structure. This can save thousands of TerraByte. Another major industry is chat bot where user input query in his on style which the system find difficult to understand what the user actually meant. The project has also application where we need to compare whether two statements are similar or not. In academic institutions, the written exam is being evaluated by teachers which is a tedious task to complete such large number of papers and within the restricted time. Therefore, the teachers can set the answer for the question and when the answer by student is provided, will evaluate to find how much similar the answer is, and can provide marks accordingly. Many more are there.

Chapter 2

Literature Review

2.0.1 Quora Question Pairs Similarity Using Logistic Regression And Support Vector Machine

Quora is a platform to ask questions about different domains. These questions can be answered by the people who have good knowledge about their area. A lot of questions are posting daily on quora, but they are not identical. Here we use various machine learning models, Logistic regression models, XgBoost models, Support Vector Machine(SVM) models. Word2vec and TF-IDF algorithms are used to identify the similarity between the questions asked by the users.

They used feature engineering to compare various machine learning models. We can convert the sentences into numerical form and given them as input to the models. Thus the model is capable of finding the similarity of two sentences. With the dataset or question pair, they tested machine learning models. This model will help to reduce log-loss for getting a better result.

Accuracy :80%

2.0.2 Exploring Deep Learning In Semantic Question Matching

In Quora duplication of similar questions is the main problem. So that the user is not able to find the exact answer. This problem can be solved by Machine learning and other processing models. More than forty thousand questions in quora are pre-processed using tokenization and lemmatization. All the datasets are converted into vectors. Distance between words, normalized features, word-share ratio, skew factors, and fuzzy-wuzzy parameters are calculated.

And apply feature extraction to this dataset, it works on 300 dimensions. Data Extraction, Dataset Preprocessing, Dataset Description, Relation between two questions, Feature Extraction, Feature Engineering, Supervised Machine Learning Models, Neural Network Design are the steps used to build the model. Deep Learning and ANN model made a huge role to detect repeated questions and giving the accurate answer. After testing six algorithms they came to know that Random Forest is better. After optimizing the artificial neural network there is a better improvement in the accuracy.

Accuracy :80.74%

2.0.3 A Model Based On Dual-Layer Attention Mechanism For Semantic Matching

It is a challenging task to match question pairs with the same meaning. This paper proposed a novel model based on a dual-layer attention mechanism. Pre-processing is done to reduce redundant information. By using the models of Question Answering Convolutional Neural Network and Question Answer Bi-directional Long Short Term Memory, CNN is applied to the sentence vectors for decision-making. The matching ability of the model can be increased by the attention mechanism.

This paper proposes the Attentive Pooling-Convolutional Neural Network and AP-BiLSTM models, a structure named matching-aggregation is proposed. Using BiLSTM,

two sentences are first encoded and matched for final decision making. A natural language speculation task is done to evaluate the stability of the model. By the adoption of attention-based preprocessing, this model can identify more heavier words among the sentence and can avoid redundant data. They claim that this model will perform better when it is extended to natural language inference tasks.

Accuracy :87.4%

2.0.4 Research On Question Answering System Based On Bi-LSTM And Self-Attention Mechanism

With the emergence of AI, deep learning, more and more people began to use the deep learning method for natural language processing. This paper proposes a model with Bi-LSTM and self-attention mechanism model. Here self-attention is used to find the relationship between words of a sentence and perform softmax. Bi-LSTM is used to encode the question and answers. With the help of deep learning the model will be capable of understanding the inner meaning of each word, so that a Bidirectional Long Short Time Memory Network is used.

While processing long information sequences the length limitation can be avoided because of the self-attention mechanism. The result shows that Bi-LSTM or self-attention mechanism greatly improves the semantic information and features of each word in a sentence. Thereby it improves the accuracy and effect of the model answer. so the accuracy rate of this model was increased by 1.6 percent, and the recall rate was increased by 1.5 percent.

Accuracy:65%

Chapter 3

System Study

3.1 Modules

The system is divided into three modules :

- (1) Query Input Module
- (2) Pre-processing Module
- (3) Feature Engineering Module
- (4) Training Module
- (5) Testing Module
- (6) Output Generation Module

3.1.1 Query Input Module

The proposed model takes two questions as input from the user through a UI. Two input fields are provided, where the user can type the two questions or two statements that he/she wants to check the similarity. The user inputted sentences are then sent to the server, where the model is present.

For training the module, we use the dataset provided by Quora as part of the Quora Question Pairs competition conducted on Kaggle. The dataset is a CSV file and contains around four lakhs question pairs of different domains.

3.1.2 Pre-processing Module

The inputs received from the Query Input module first go through a pre-processing stage before given to the model. The pre-processing module contains different functions. Initially, the abbreviations and short forms are changed to elaborated forms. Then will tokenize the sentence. After that, vectors are created for each word and padding applied. Vectors are a sequence of integers assigned to each word. These are given to the model as input.

3.1.3 Feature Engineering Module

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data ^[1]. In our model, we also use feature engineering as the input to the model. Two categories of feature engineering are used, text classification and NLP feature engineering. Text classification is the problem of assigning categories to text data according to its content ^[1]. This includes stop-words removal, Stemming and Lemmatization. NLP feature engineering mainly contains different distance calculations.

3.1.4 Training Module

The main task in the training module is to train the proposed model. Two main layers are present in the model, i.e. CNN and Bi-LSTM. CNN is used to capture crucial feature information, while the BiLSTM network is used to perform semantic analysis of the entire sentence ^[2]. The model accepts three inputs, they are the vector form of two questions

from the training dataset and the features of each question. Then these inputs are given to the CNN-BiLSTM layer to train the model. The model predicts output of range between 0 and 1.

3.1.5 Testing Module

The trained model which predicts the amount of similarity between the two questions inputted by the user is already saved. In the testing module, the saved model is loaded. It takes three inputs, the questions in vector format and their features. The predicted similarity is in the range between 0 and 1.

3.1.6 Output Generation Module

The Web application is designed on the Flask framework and has a homepage with two user input box. The user input the question or statement which he needs to check for similarity. On submitting the query, the model will be loaded. The statement/question will undergo pre-processing and word2vec formation and also feature engineering is done on the input. These will be loaded into model and the final predicted similarity will be between 0 and 1. This is converted to percentage and is returned to home template. This is the final result that we expected.

Chapter 4

System Design

4.1 Use Case Diagram

The use case diagram, Figure 4.1, for the proposed model described as follows: two actors are present in the model, the Data Analyst and the User. The diagram describes the external things that interact with the part of the system. Data Analysts do the data collection, data analysis feature engineering of the datasets. Data Analysts train and evaluate the model. Also, make parameter tuning and output prediction on the input. Then convert the result into a two-digit number. The User actor gives two queries as input and views the predicted result.

4.2 Sequence Diagram

The sequential diagram, Figure 4.2, describes the predicted model as follows: Initially, the user gives two questions as the input to the system. The system creates a copy of the questions and converts them into a sequence of integers (vectors). The questions is then given to the feature engineering module where NLP features and text classification features are generated as output. Its output is given to the trained model together with

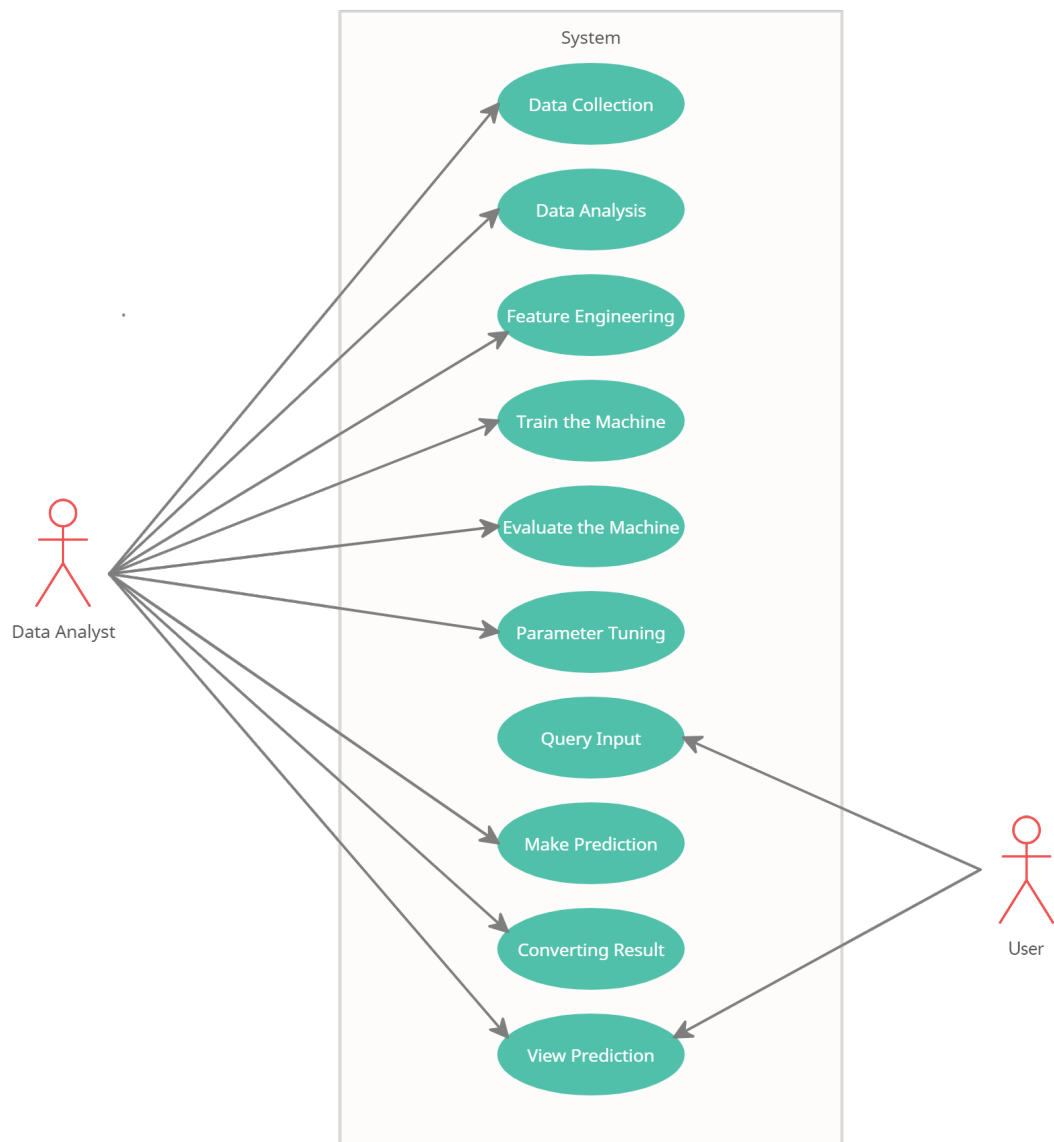


Figure 4.1: Use Case Diagram

the vectors. The predicted output is then rounded to a two-digit number by multiplying with hundred and gives as output to the user.

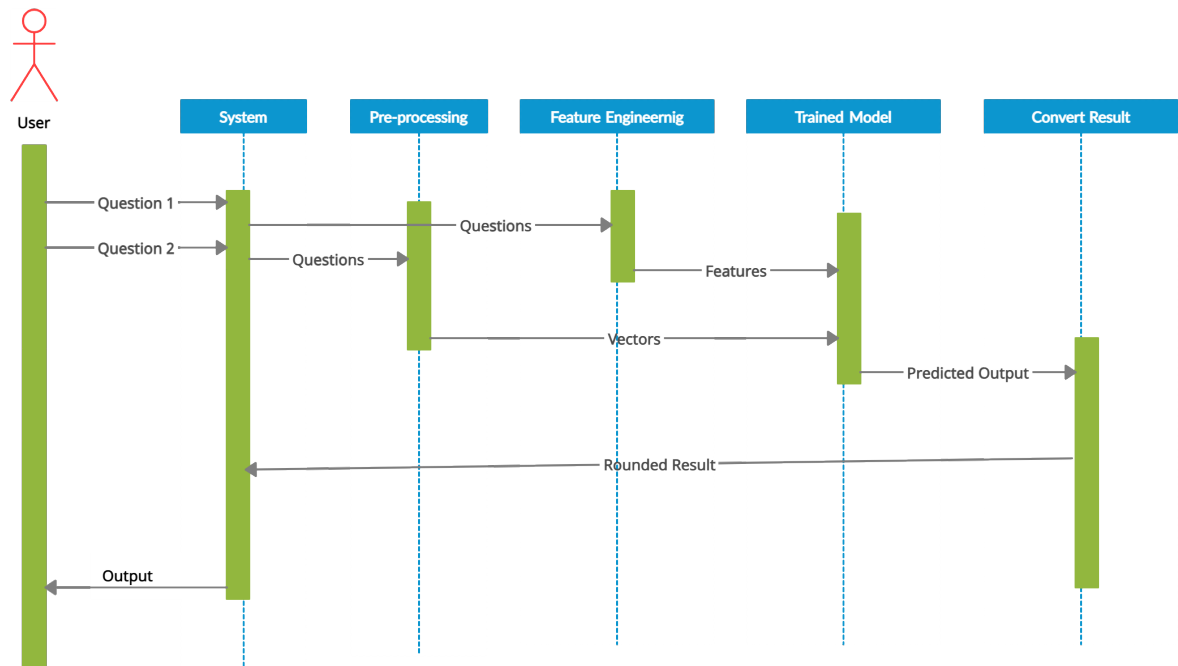


Figure 4.2: Sequence diagram

4.3 Architectural Diagram

4.3.1 Training Architecture

Figure 4.3 below represents the architectural diagram of training module. The Quora Question Pair dataset is used in our project. In the training process, the dataset is used for pre-processing the question pairs and for feature engineering. In pre-processing, the question pair may have some abbreviation that system didn't understand. This will be converted to known format. Followed by tokenization and Text-to-Sequence where each word will have an integer value. On that, padding is done to make it of equal length. In feature engineering section, the dataset will be engineered to generate different features. Then stop words are removed and distance is measured.

The output from both pre-processing and feature engineering are input to the model we developed. The model is employed with CNN and Bi-LSTM. CNN is an excellent deep learning algorithm for feature extraction from a visual imagery or a statement. Bi-LSTM

is a sequence processing module. The model will generate the corresponding output.

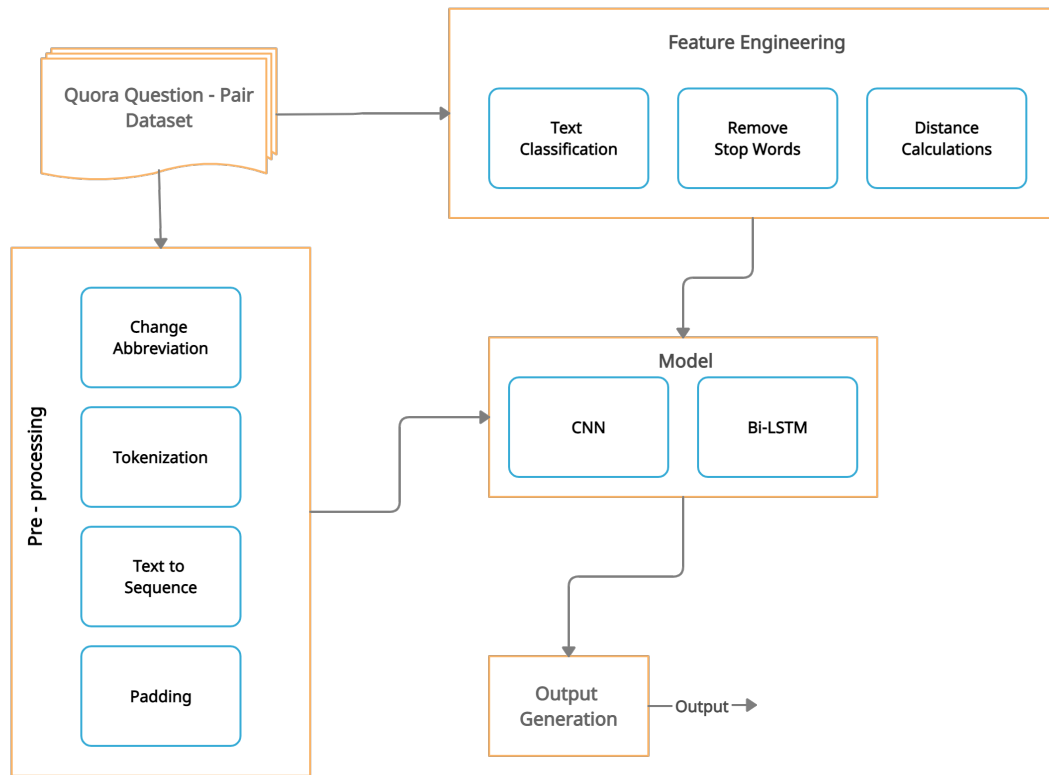


Figure 4.3: Training Architectural diagram

4.3.2 Testing Architecture

Figure 4.4 shows the architecture of testing process. In this process, we have input query which is fetched to pre-processing and tokenization which is subjected to trained model. Similarly, the input query is subjected to feature engineering and certain features are generated. The result along with the feature engineering result of training set will be normalized and is given to trained model. The pre-processed input query along with normalized result will be used by trained model to produce similarity probability. This is then rounded to produce output.

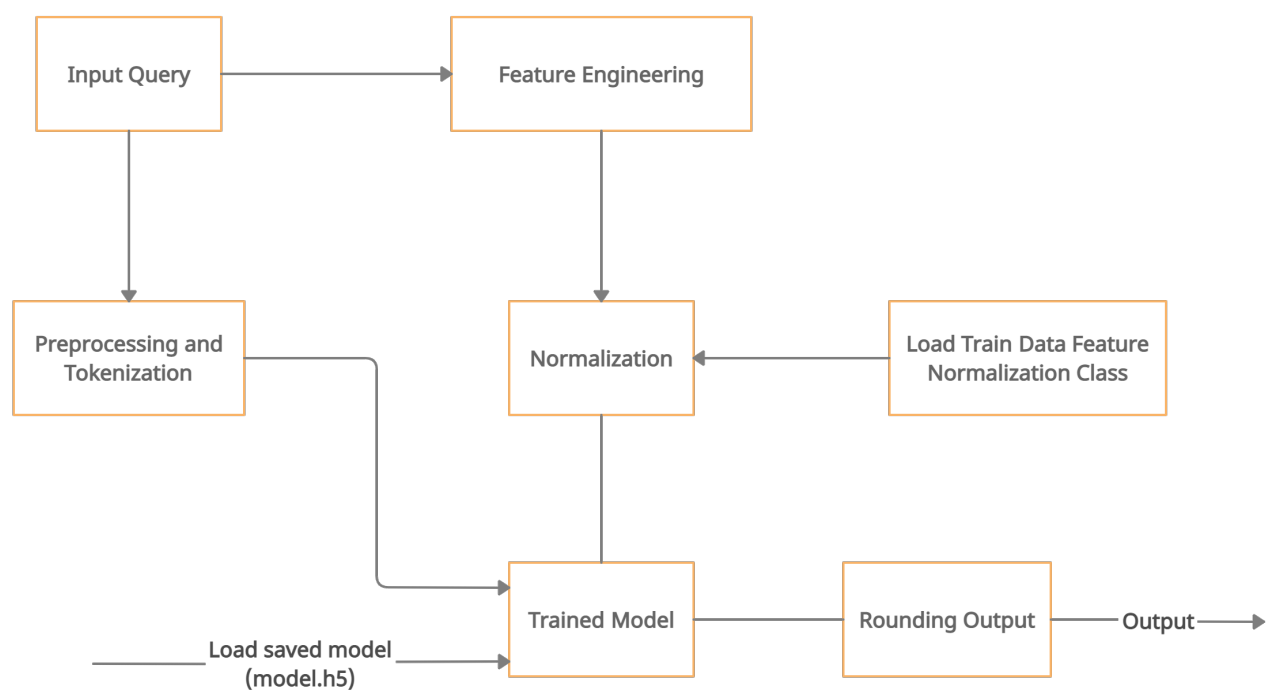


Figure 4.4: Testing Architectural diagram

Chapter 5

Project Design and Implementation

5.1 Design

The project design comprises of the processes involved in the overall implementation of the project. The project design involves the following stages:

5.1.1 Data Collection

The data collection is the primary step involved in the project. The datasets used for this project are provided by the Quora Question Pair competition in Kaggle. The file contains two CSV format files, the train.csv and the test.csv. The train.csv file contains six attributes like *id*, *qid1*, *qid2*, *question1*, *question2* and *is_duplicate*. The test.csv file contains three attributes like *test_id*, *question1* and *question2*. For training purpose, the train.csv file is loaded into DataFrame which contain more than 4 lakhs questions over different domains. Similarly, for testing purpose, the test.csv file is loaded into DataFrame which contain more than 15 lakhs questions but here we use only five hundred questions. We cannot pass the texts directly to the CNN model to predict. So we need to convert the text into a vector format. For this, glove.840B.300d.txt is used which contains 840

billion words of three hundred dimensions.

5.1.2 Data Preprocessing

The data from csv files are loaded into DataFrame for Data Preprocessing. In this phase, we convert the short form of the words into the original form, remove stopwords, remove null values, make the data suitable for Feature Engineering, CNN and BiLSTM.

5.1.3 Feature Engineering

Feature Engineering is a combination of Text Classification and NLP features which extract forty-two features from each input. This layer makes the model more efficient and also increases the accuracy by an additional eight percentage and also plays a major role in reducing the log loss of the input.

5.1.4 Model Creation

Develop a model to predict whether two input questions are the same or not. This phase consists of two main parts:-

Convolutional Neural Network (CNN)

Convolutional Neural Network is a deep learning methodology that plays a major role in this project. The CNN is a neural network that mimics the exact working of human brain neurons. The CNN is used to capture the crucial features of the input questions. Here we use two one-dimensional CNN layers. Conv1D is used because the words are one-dimensional array as compared to an image (2D array).

Bidirectional Long Short Term Memory (BiLSTM)

Bidirectional Long Short Term Memory is a memory cell in RNN which is more efficient in the case of text data prediction. In our project, the BiLSTM is used to extract the semantic features or understand the meaning of the input questions. Instead of LSTM, BiLSTM is used here because the BiLSTM can move in both forward and backward direction which effectively increase the amount of information available in the network. We fused CNN and BiLSTM to extract and represent important textual information and contextual information.

5.1.5 Output Generation

In the output generation phase, we request two questions from the user then and using our model check their semantics same or not. Then display the result to the user, by how much two questions are similar.

5.2 Implementation

GPU enabled TensorFlow backend is used in the project. Matplotlib is used to visualize the data. FuzzyWuzzy and Distance libraries are used to calculate different distances function. Wordcloud and nltk libraries are used to preprocess the inputs. Python is the programming language used for developing the model. Google Colab with GPU is used as IDE. Flask web framework is used to integrate the model to UI

5.2.1 Data Preprocessing

The two datasets train.csv and test.csv files are loaded into two DataFrames and is print. In the training file, 36.9% are duplicated questions and 63.1% are non duplicated ques-

tions. The initial step of this phase is to convert the short-form words into their original form. i.e isn't to is not, haven't to have not, likewise for both the train and test questions. Then tokenize each word to text-to-sequence i.e. sequences of integers (vectors). In our case, we have more than eight thousand unique words. Then we add padding to each word with sixty as the word length. In addition to the above process, we added a new feature to find the frequencies of questions and take the intersection of them.

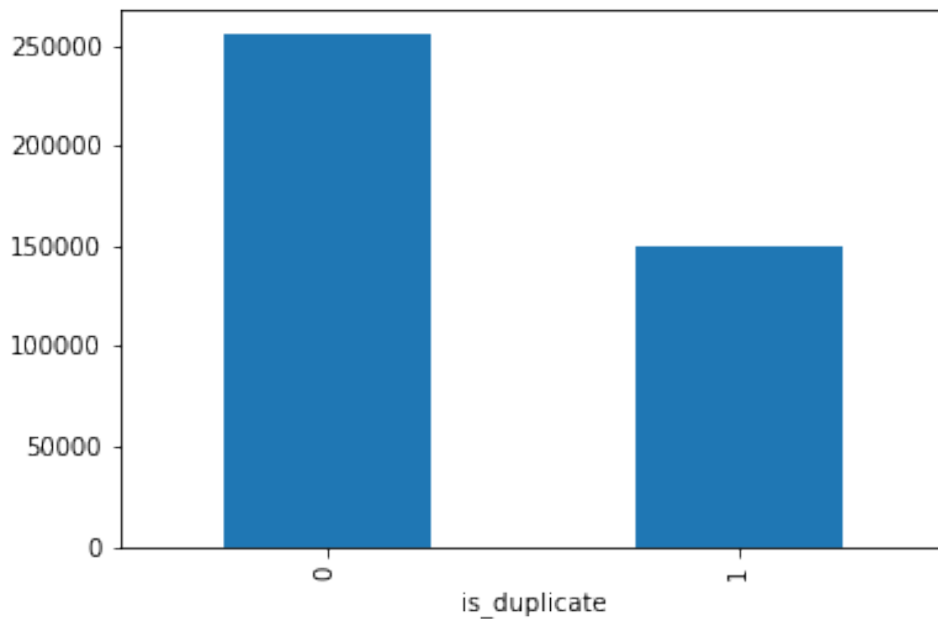


Figure 5.1: Training Data Graph

5.2.2 Feature Engineering

For creating a better and more efficient model we add a new methodology Feature Engineering. In Feature Engineering two methods are used Text Classification and NLP features. From these forty-two new features were found.

Text Classification

In the Text Classification, we extract 19 features from two input questions. Features are:- *ques1_len* - is the length of question 1, *ques2_len* - length of the question, *len_diff* -

the difference between the length of two questions, *q1_word_len* - question 1 word length, *q2_word_len* - question 2 word length, *words_diff* - the difference in the length of words of two questions, *q1_caps_count* - question 1 capital word count, *q2_caps_count* - question 2 capital word count, *caps_diff* - the difference between count of capital words of two questions, *q1_char_len* - length of characters of question 1, *q2_char_len* - length of characters of question 2, *diff_char_len* - the difference between characters of two questions, *avg_word_len1* - average word count of question 1 between two questions, *avg_word_len2* - average word count of question 2 between two questions, *diff_avg_word* - the difference between average word count of two questions, *common_word* - the intersection between two questions, *total_word* - total word count, *word_share* - the intersection of two questions by total length of two questions, *share_2_gram* - word 2 gram of two questions. Figure 5.2 is an example of the Text Classification feature extraction of two questions.

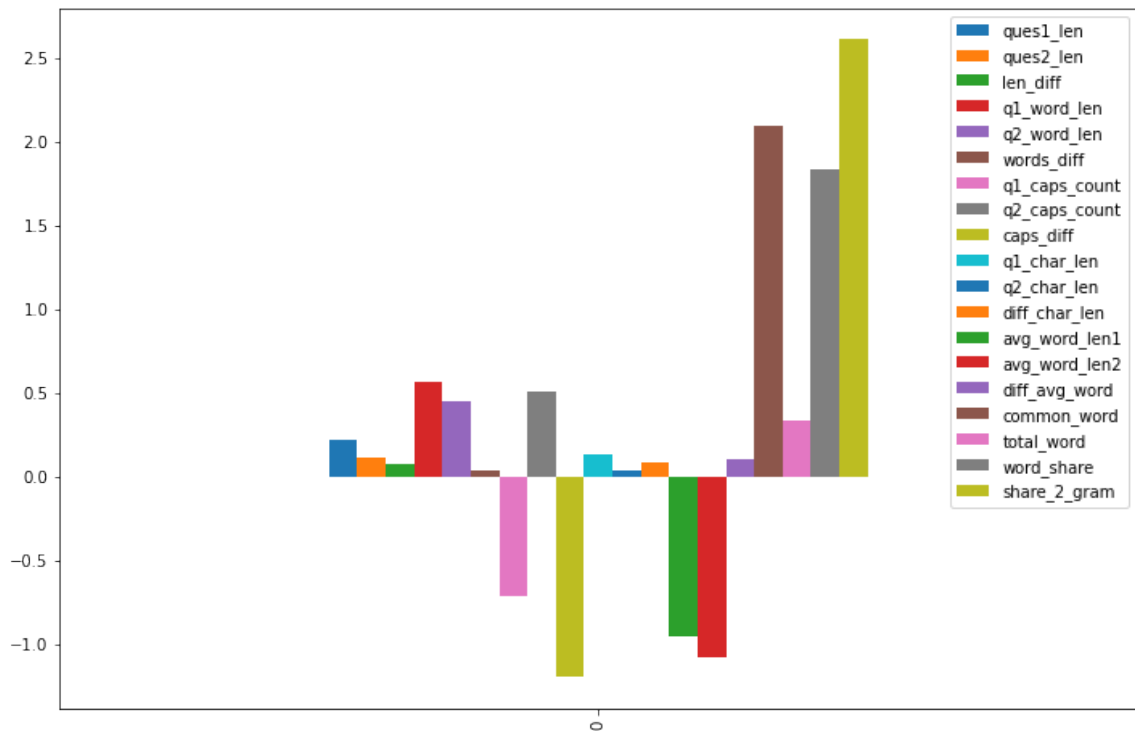


Figure 5.2: Text Classification Bar Graph

NLP Features

In NLP Feature Engineering, we extract new 21 features from two questions. Features are *cwc_min* - count of common words other than stop words in two questions by a minimum of the length of two questions other than stop words, *cwc_max* - count of common words only in stop words in two questions by a maximum of the length of two questions other than stop words, *csc_min* - count of common words only in stop words of two questions by a minimum of the length of two questions only in stop words, *csc_max* - count of common words only in stop words of two questions by a maximum of the length of two questions only in stop words, *ctc_min* - count of common words in two questions by a minimum length of two questions, *ctc_max* - count of common words in two questions by a maximum length of two questions, *last_word_eq* - check whether the last word of two questions are same or not, *first_word_eq* - check whether the first word of two questions are same or not, *abs_len_diff* - absolute length difference of two questions, *mean_len* - mean of length of two questions, *token_set_ratio* - take fuzzy set ratio of two questions, *token_sort_ratio* - take fuzzy sort ratio of two questions, *fuzz_ratio* - fuzzy Qratio of two questions, *fuzz_partial_ratio* - fuzzy partial ratio of two questions, *longest_substr_ratio* - fuzzy longest substring ratio of two questions, *word_mover_dist* - wmdistance of two questions, *cosine_dist* - cosine distance between two questions, *cityblock_dist* - cityblock distance between two questions, *canberra_dist* - canberra distance between two questions, *euclidean_dist* - euclidean distance between two questions, *minkowski_dist* - minkowski distance between two questions. Figure 5.3 is an example of the NLP feature extraction of two questions.

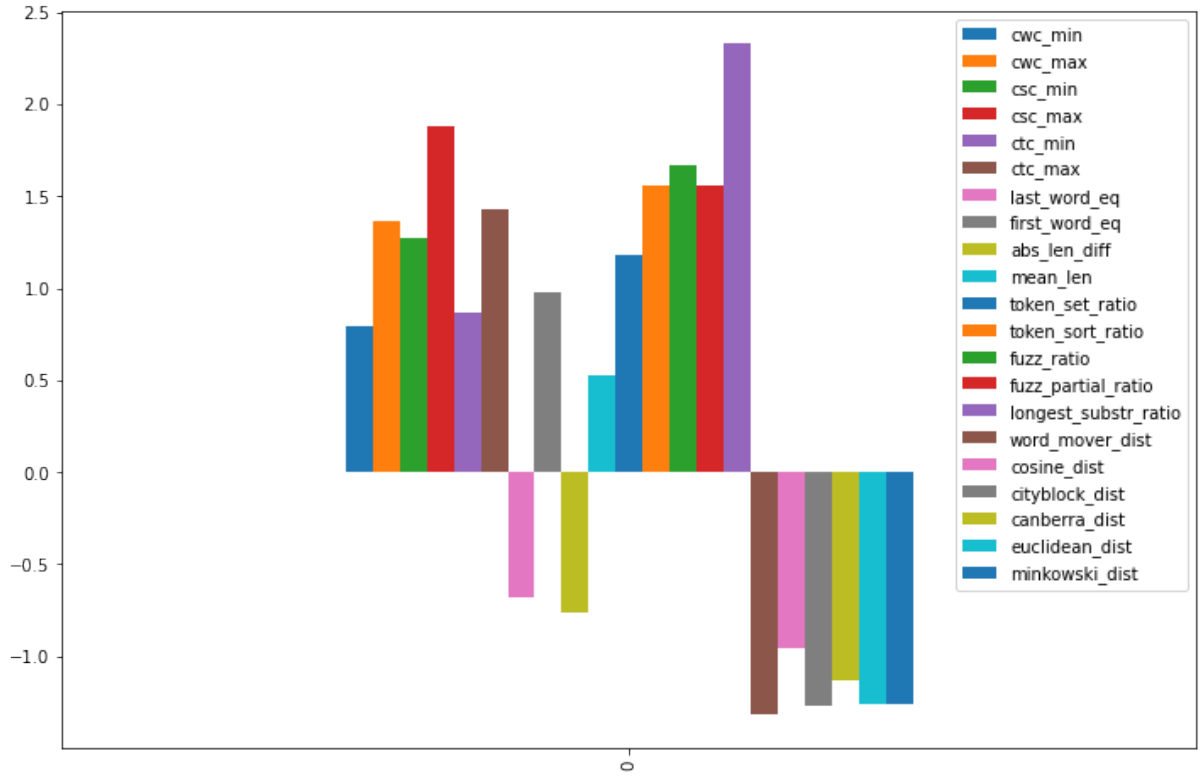


Figure 5.3: NLP Features Bar Graph

5.2.3 Model Creation

Our model is the most efficient and the most effective for question similarity calculation. We fuse both CNN and BiLSTM to get accurate and precise output. In our model three inputs, we have question 1, question 2, and feature engineering values of two questions. The two input questions are passed to the embedding layer and copy output into three. The two copies are passed into the Conv1D layer and one copy is passed to BiLSTM. The output of the two Conv1D layers is passed into the GlobalMaxPool layer to reduce the dimension into two and then concatenate both. At the same time, the output of BiLSTM and dense output of feature engineering are concatenated. Then the output of both layers is separately passed to the BatchNormalization layer, then to the Dropout layer and followed by that, to the Dense layer. Now we concatenate both of them and then passed to series of different layers and finally predict the output using a sigmoid function. Figure 5.4 shows visualization of the proposed model.



Figure 5.4: Model Generation

5.2.4 Output Generation

The output generation phase mainly focuses only on UI and UX. This phase is developed using HTML, CSS programming languages, and Flask frameworks. We develop a webpage to interact with the user and host the webpage as a localhost on our system. When the

user inputs two questions in the input box, we pass the questions to our backend system. After completing the backend process, our model returns the semantic similarity between two questions. Then the output display on the webpage as a percentage value.

Chapter 6

Experimental Results

6.1 Training Data

Question 1	Question 2	Duplicate
What is the step-by-step guide to investing in the share market in India?	What is the step-by-step guide to investing in share market?	0
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
How do I read and find my YouTube comments?	How can I see all my Youtube comments?	1
What can make Physics easy to learn?	How can you make physics easy to learn?	1
What does manipulation mean?	What does manipulation means?	1

Figure 6.1: Training Data

Figure 6.1 shows the sample of our training dataset. The training dataset contains more than four lakh questions. The data loaded into DataFrame are kept for data preprocessing. Null columns are removed. We have done all preprocessing like tokenizing, word embedding, padding, feature engineering on the DataFrame and add more features to it. The preprocessed data were passed into our model to train with four epochs. The model fuses with Conventional Neural Network and Bidirectional Long Short Term Memory.

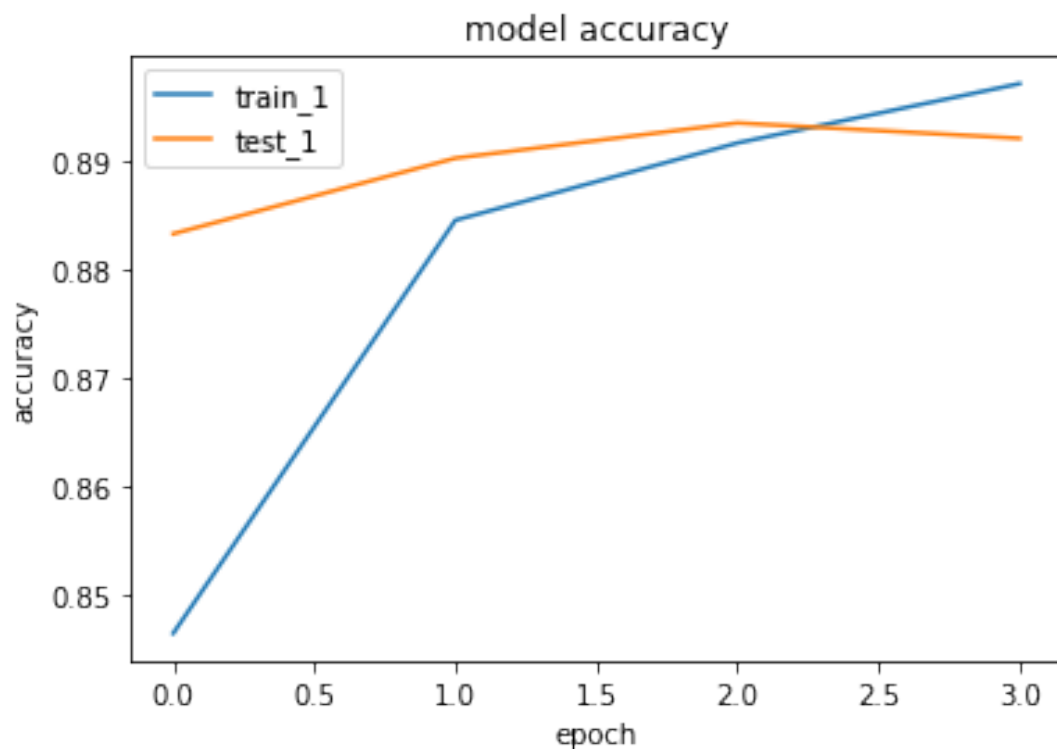


Figure 6.2: Accuracy Graph

6.2 Training Result

Validation Accuracy

With CNN, BiLSTM, and Feature Engineering our model has an accuracy of 90% and validation accuracy of 89%. Without Feature Engineering, our model have only 79% accuracy. With feature engineering, we can increase an additional 10% accuracy to our model. The model train with four epochs, more than this will result in overfitting the model. Figure 6.2 shows the accuracy graph.

Log-Loss

Figure 6.3, shows the Log-Loss of the model. After four epochs 0.22 is the loss and 0.23 is the validation loss of the data. The loss function is *binary cross-entropy* and the optimizer

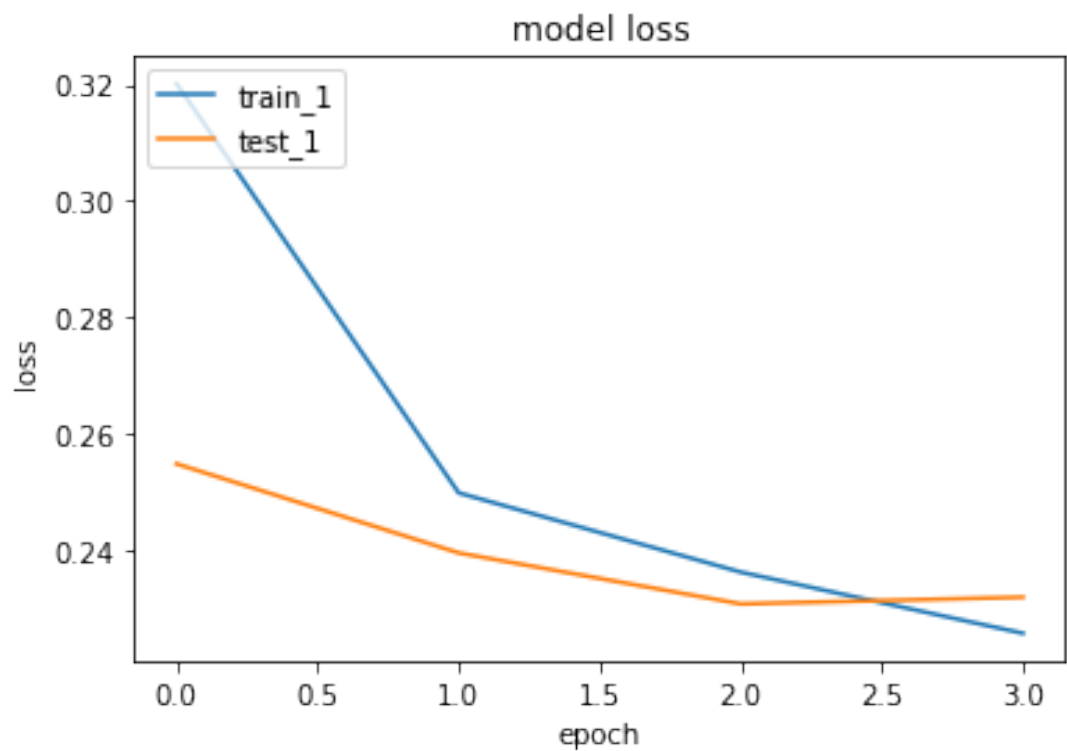


Figure 6.3: Log-Loss Graph

is *adam*. Feature Engineering also plays a major role in reducing the loss of the model by 0.2 from 0.5.

Confusion Matrix

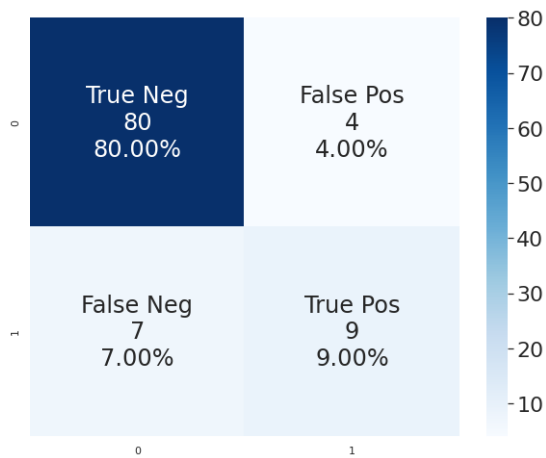


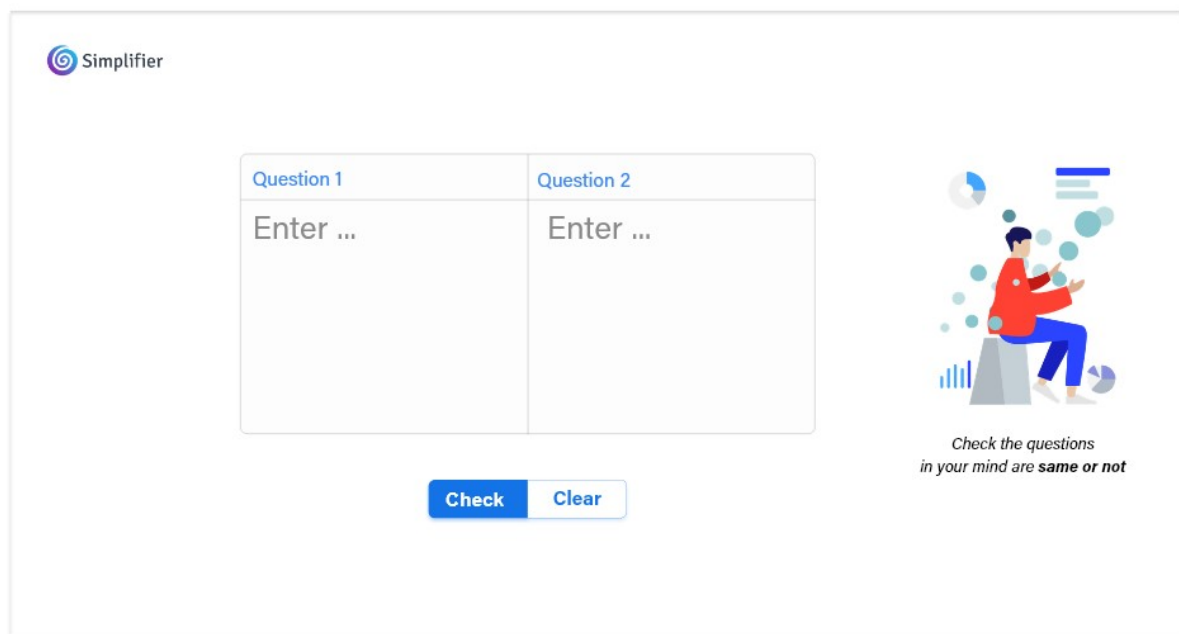
Figure 6.4: Confusion Matrix

We take one hundred random questions from the testing data which is not trained during the training phase and predict the output using our model. When observing the predicted data we found that out of one hundred 89 questions are correctly predicted.

Evaluation Result

S.No	Methods	Accuracy
1	Using Logistic Regression and SVM	80%
2	Using CNN	82%
3	Using attention layer	86%
4	Our Proposed Model	89%

Figure 6.5: Evaluation Result



The screenshot shows the home page of an application named 'Simplifier'. It has a clean, modern interface with a light gray background. In the top left corner, there is a logo consisting of a purple swirl icon followed by the word 'Simplifier'. The main content area contains two side-by-side input fields, each with a blue header 'Question 1' and 'Question 2' respectively, and a placeholder text 'Enter ...'. Below these fields are two buttons: a blue 'Check' button and a light blue 'Clear' button. On the right side, there is a colorful illustration of a person in a red jacket and blue pants sitting on a gray rock, surrounded by various data visualization icons like pie charts, bar charts, and circles. Below the illustration, there is a text prompt: 'Check the questions in your mind are same or not'.

Figure 6.6: Home Page

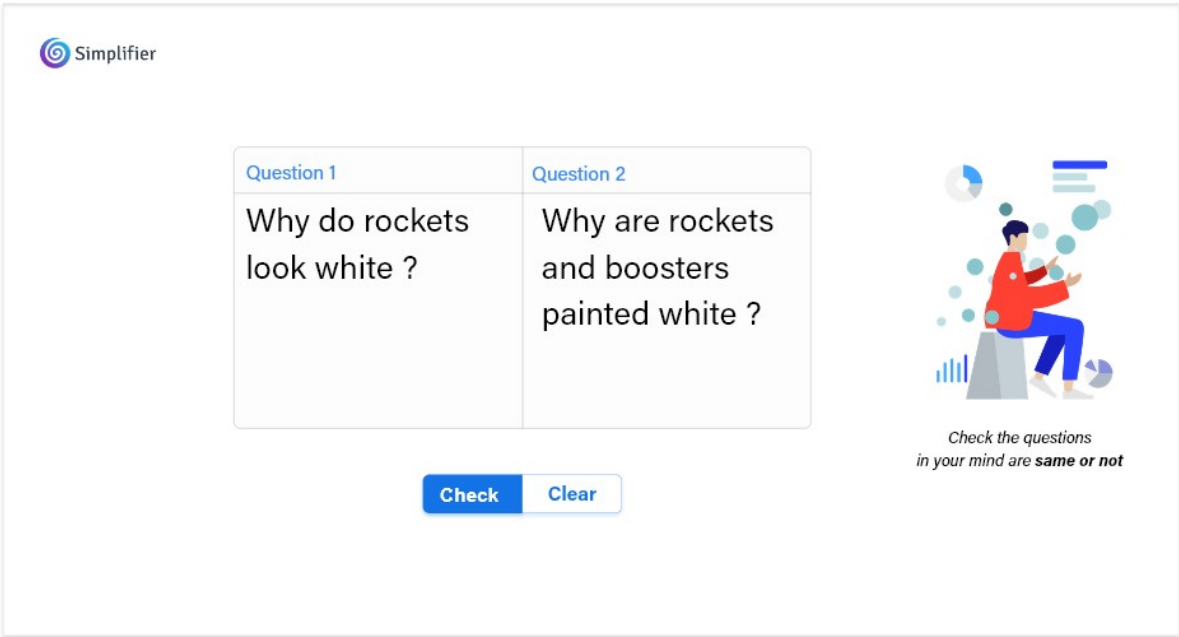


Figure 6.7: When Questions Entered

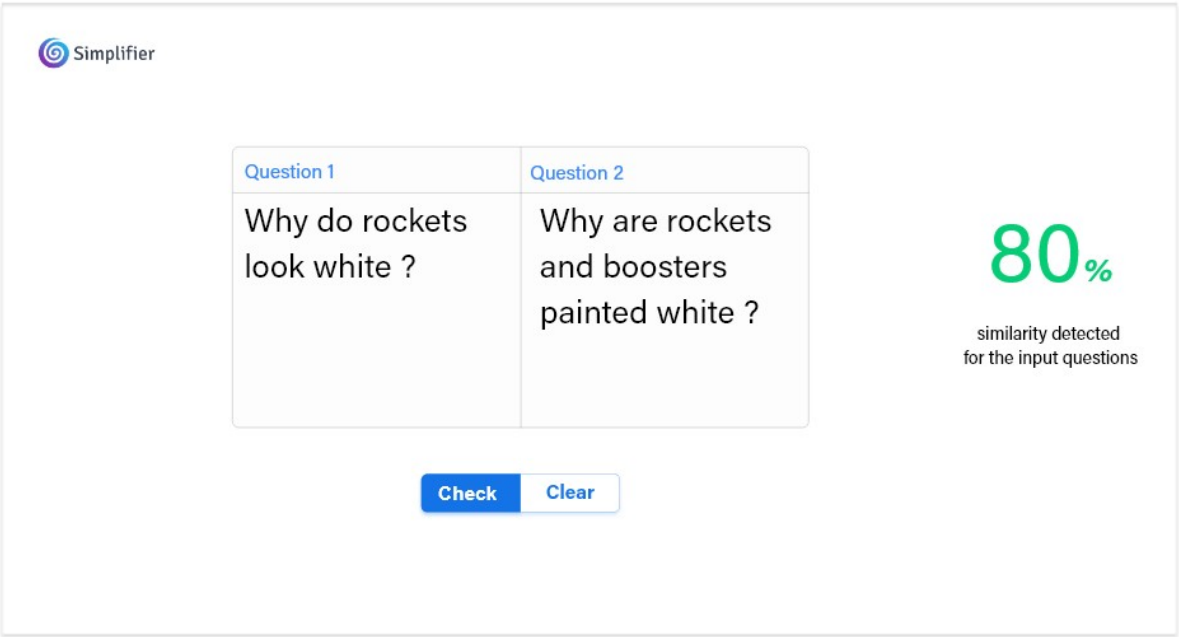


Figure 6.8: Similarity Between Questions

Conclusion

Understanding the natural language is something inevitable in current scenario. This can break the barrier between a human being and computer. The problem we faced so long is to make the computer familiar with our language. Our project has succeeded in making this task so easier. Our model intelligently understood the meaning of each statements using Deep Learning and matched the similarity between two questions or statements. Employing our project in arious industries will help to leap ahead in business and market much better.

After the experiment, we come to acknowledge that our model has achieved a prediction accuracy of **89%** and a reduced loss rate of mere **0.22**. This surely will prove that our model is far better than any other product available in the market

The project can be an asset to giant search engine industry like Google, Yahoo... and chat bots where user has his/her own style in inputting the query. Also academic institutions has a great potential with this project. Country like India has the education system where written exam is carried out. Under such cases, they found difficult to evaluate the answer script of all these students and that too, in the prescribed time. This can cause so much delay in valuation and result publish. Using the project can definitely a great helping hand to such problems. There exist many more industries which can adopt this project to make exciting leap in their day-to-day activities.

References

- [1] <https://towardsdatascience.com/text-analysis-feature-engineering-with-nlp-502d6ea9225d> - "*Text Analysis Feature Engineering with NLP*"
- [2] L. -Q. Cai, M. Wei, S. -T. Zhou and X. Yan, "*Intelligent Question Answering in Restricted Domains Using Deep Learning and Question Pair Matching*," in IEEE Access, vol. 8, pp. 32922-32934, 2020, doi: 10.1109.
- [3] *Quora Question Pair dataset from Kaggle* (<https://www.kaggle.com/quora/question-pairs-dataset>)
- [4] *Glove - Global Vectors for Word Representation* (<https://nlp.stanford.edu/projects/glove>)
- [5] Dr. P V Rama Raju, G. Naga Raju, N. Nikhil, M.Hemanth Gupta, Chandan Akella, S.Kumara Siddarth "*Quora Question Pairs Similarity Using Logistic Regression and Support Vector Machine*"
- [6] H. Xiang and J. Gu, "Research on Question Answering System Based on Bi-LSTM and Self-attention Mechanism," 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), 2020, pp. 726-730, doi: 10.1109
- [7] A. Dhakal, A. Poudel, S. Pandey, S. Gaire and H. P. Baral, "*Exploring Deep Learning in Semantic Question Matching*," 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), 2018, pp. 86-91, doi: 10.1109.

- [8] T. Liu, S. Yu, B. Xu, and H. Yin, "Recurrent networks with attention and convolutional networks for sentence representation and classification," *Appl. Intell.*, vol. 48, no. 10, pp. 3797–3806, Oct. 2018.
- [9] L. Cai, S. Zhou, X. Yan, and R. Yuan, "A stacked BiLSTM neural network based on coattention mechanism for question answering," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–12, Aug. 2019.
- [10] Y. Xiang, Q. Chen, X. Wang, and Y. Qin, "Answer selection in community question answering via attentive neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 4, pp. 505–509, Apr. 2017.
