Quora Question Pair Similarity Predictions

Mohammad Talib
IIIT Delhi
Okhla Industrial Estate, Phase III
New Delhi, India
talib18245@iiitd.ac.in

Bassam Pervez Shamsi
IIIT Delhi
Okhla Industrial Estate, Phase III
New Delhi, India
bassam19032@iiitd.ac.in

1. INTRODUCTION

Quora is a spot to acquire and share information—about anything. It's a stage to pose inquiries and associate with individuals who contribute one of a kind bits of knowledge and quality replies. This engages individuals to gain from one another and to all the more likely comprehend the world.

More than 100 million individuals visit Quora consistently, so it's nothing unexpected that many individuals pose likewise phrased inquiries. Various inquiries with a similar expectation can make searchers invest more energy tracking down the most intelligent solution to their inquiry, and cause authors to feel they need to answer numerous renditions of a similar inquiry. Quora esteems sanctioned inquiries since they give a superior encounter to dynamic searchers and journalists, and proposition more worth to both of these gatherings in the long haul. [1]

1.1.1 Problem Statement

Given a pair of questions, our task is to determine if the questions have similar intent.

1.1 Motivation

Quora is an online platform where people ask questions across a wide variety of topics. Identifying question similarity prevents users from asking duplicate questions, it also saves time for domain experts. Users will be able to see answers to questions with similar intent, and won't have to wait for answers.

1.2 Online Deployment link

https://nlp-quora-similarity-demo.herokuapp.com/

1.3 Project Pipeline Summary

1. We acquired the labelled dataset from kaggle. After acquiring the data we performed the preprocessing steps, as well as exploratory data analysis.
2. We then performed feature extraction.
3. After feature extraction, we experimented with various models, and did thorough analysis of their performance.

   We tested accuracy, F1 scores, precision and recall for all the models which we trained.
4. Finally, we selected our best performing model and deployed it on heroku using the Django Web Framework.

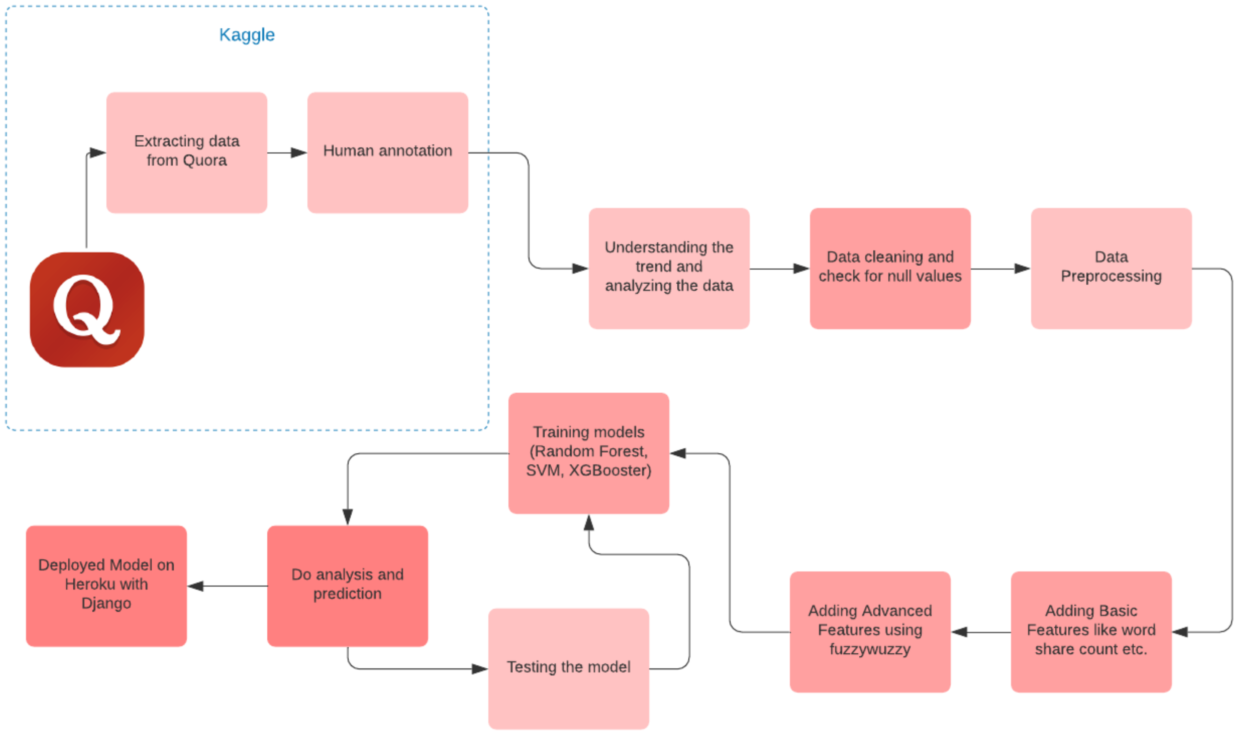

Kaggle
Extracting data from Quora
Human annotation
Understanding the trend and analyzing the data
Data cleaning and check for null values
Data Preprocessing
Adding Basic Features like word share count etc.
Adding Advanced Features using fuzzywuzzy
Training models (Random Forest, SVM, XGBooster)
Testing the model
Do analysis and prediction
Deployed Model on Heroku with Django

Figure 1 : Project Pipeline Summary

1.4 Related Work

LSTMs were incorporated into many models for question pair classification. Word and sentence embeddings were utilized a lot as well.

The first place solution (on Kaggle) for the problem used embeddings such as Word2Vec, Glove, FastText, Doc2Vec, Sent2Vec. [1]

It used two main architectures for their Neural Networks, Siamese and Attention Neural Networks.

Their model consisted of 4 layers, on the first layer, they had 300 models, which were a combination of neural networks, classical algorithms and Scikit-learn classification algorithms.

Second layer consisted of all the predictions from the first layer, along with all the input features, as well as a hidden layer.

Third Layer consisted of 2 Linear models.

and finally, the fourth layer consisted of a 55/45 Blend.

Other submissions mainly used SVM, Random forest models, and LSTMs and other deep learning techniques.

2. MATERIALS AND METHODS

2.1 Data

2.1.1 Data Collection

We collected the labelled dataset from Kaggle. And used it in our model for prediction. [3]

2.1.2 Data Overview

Data will be collected in a file train.csv. It contains rows having a pair of questions and some more attributes like question id of first question i.s qid1, question id of second question i.s qid2. And there are binary values in the last column in each row showing if the questions have similar intent. The label in the last column that we are trying to predict - whether the two questions are duplicates of each other. The size of training data is approximately 60 MB.

2.1.3 Data Preprocessing

We followed the below steps to pre-process the data and reduce the redundancy of the data:

- Tokenization
- Removal of all null values, allowing only standard English words.
- Removal of NaN values, filling those values with empty string.

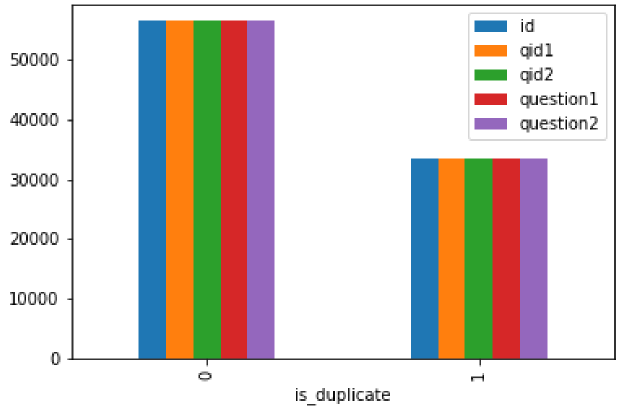
2.2 Methodology

2.2.1 Dataset Statistical Analysis

We do statistical analysis of the whole dataset to determine the statistics, content and trends of the quora dataset by common computational exploratory processes and provide simple answers.

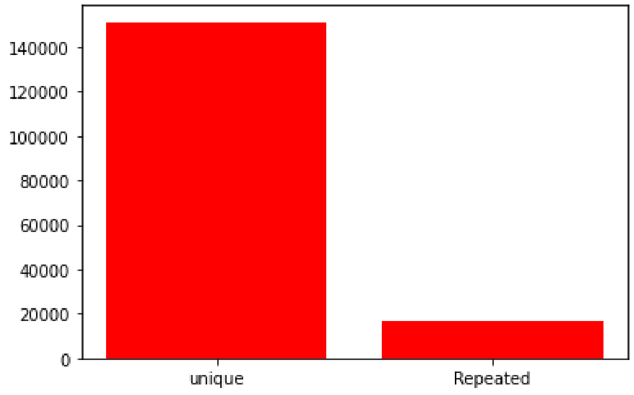
2.2.1.1 Some Basic Statistics

We determine the content by grouping our whole dataset into two classes similar and non-similar. There are 62.9% question pairs which are not similar to each other. And 37.21 % question pairs which are similar to each other.



2.2.1.2 Other statistics

In the dataset, there are 16847 questions (11.15 %) unique questions that are repeated. There are 151098 unique questions present in the dataset which will be further used to train the model. Based on this, we also calculated the frequency of a single question which is repeated the maximum number of times in the dataset.



2.2.2 Extracting Basic Features from the Dataset

We do quantitative analysis of all datas by associating them with new features like length of questions, number of words in each pair of questions, number of unique words in each pair of questions and total number of words present in each pair of questions, word share that is calculated by dividing common words by total words present in questions. For such categorization we used the apply function which can be applied to whole columns or rows.

2.2.2.1 Analysis of some of the extracted features

We determine some statistics using basic feature extraction like how many questions are of minimum length and minimum length of questions from any of the questions from a pair.

Minimum length of the questions in question 1 : 1

Minimum length of the questions in question 2 : 1

Number of Questions with minimum length [question 1] : 14

Number of Questions with minimum length [question 2] : 5

We do analysis on the basis with basic features, it is not much significant for a question pair with high correlation features and metrics. Then we moved on for advanced feature extraction for the dataset. [3]

2.2.3 Addition of Advanced Features for the Data

We have seen that basic features are not enough for predictions as it was just a simple mathematical set of features. And We would have no trouble in picking up identical questions having similar intent while we using basic features. First we use statistical tests to establish the significance of fuzzy to have a statistical significance on the sentence similarity.[3] Then we planned to use statistical features of Fuzzy scores generated to train machine learning prediction models. We planned to deploy a system/dashboard on heroku using the Django Web Framework.

2.2.3.1 Fuzzy Based Predictions

We added advanced features like fuzz ratio, fuzz partial ratio, token sort ratio and token set ratio. It helped us to predict the question queries more efficiently than basic features do. The fuzz ratio tells us how much percentage these two questions are similar, measured with edit distance when we convert it into string format. And fuzz partial ratio checks whether two strings are of noticeably different lengths, we are getting the score of the best matching lowest length substring. [3]

2.3 Experiments

2.3.1 Training Models

Before training the models, We used train test split to split our data into training and test datasets. We set train_size as 0.75 and random state as 1, which implies our test data size is 25 percent (including X_test and y_test) of the whole training data and the remaining 75 percent data (including X_train and y_train) is used to train the model.

2.3.1.1 Using Random Forest Classifier

A random forest is a meta assessor that fits various choice tree classifiers on different samples of the dataset and utilizations averaging to work on the prescient precision and command over-fitting. We used a random forest classifier with 100 of decision trees and random state as 1. We saved our model as a pickled model. So that we can load back later when we want it.[4]

2.3.1.2 Using XGB Classifier

XGB assembles an added substance model in a forward stagewise style; it considers the advancement of subjective differentiable misfortune capacities. In each stage n_classes_ relapse trees fit on the negative angle of the binomial or multinomial abnormality misfortune work. Twofold arrangement is an uncommon situation where just a solitary relapse tree is initiated. We used a XGB forest classifier. We saved our model as a pickled model. So that we can load back later when we want it.[5]

3. RESULT AND EVALUATION

3.1 Results

After training our random forest model, we test the score for it. And It was giving an accuracy of 99.87% . As we set train_size as 0.75 and random state as 1, which implies our test data size is 25 percent (including X_test and y_test) of the whole training data and is used for predicting values. We stored the predictions in y_pred and used them for further analysis of the model.We saved both models as pickled models. So that we can load back later when we want them for further predictions. Similarly, we set train_size as 0.75 and random state as 1, which implies our test data size is 25 percent (including X_test and y_test) of the whole training data when we used XGBClassifier for predictions. And It was giving an accuracy of 71.88% .

3.2 Analysis of models

We wanted to do some analysis of the performance of each of the models. We generated confusion matrices for our models which we trained. We tested accuracy, F1 scores, precision and recall for all the models as well as we reported classification reports. The Random forest model has better performance than other models as It has good precision, recall, F1 score and accuracy. And after that we planned to select our best performing model, a random forest model and deployed it on heroku using the Django Web Framework.

4. ACKNOWLEDGMENTS

Our thanks to Md. Shad Akhtar for providing us with this golden opportunity to undertake this project and for this constant support and motivation in the course.

5. REFERENCES

[1] Quora Question Pairs winning solution post

[2] Quora Question Pairs (Kaggle challenge)

[3] (FuzzyWuzzy: Fuzzy String Matching in Python - ChairNerd, 2021)

[4] (sklearn.ensemble.RandomForestClassifier, 2021)

[5] (sklearn.ensemble.GradientBoostingClassifier, 2021)

---xxx---