Quora Question Pair Similarity Detection

Abhuday Tiwari
IIIT Delhi
MT22005
abhuday22005@iitd.ac.in

Ayush Agarwal
IIIT Delhi
MT22095
ayush22095@iitd.ac.in

Nikhilesh Verhwani
IIIT Delhi
MT22114
nikhilesh22114@iitd.ac.in

Ujjwal Garg
IIIT Delhi
MT22095
ujjwal22085@iitd.ac.in

1 Introduction

Quora is globally well known question-answering platform used by millions worldwide over the internet. Same questions duplicated by different instances on the platform can lead to inefficient and poorer experience for both answer seekers and writers. It would lead to seekers taking more time to find the most relevant answer while writers would experience redundancy and repetitiveness in their job. To offer more value to both parties in long-term, the platform needs to maintain unique questions' database. The task is to identify whether a given pair of questions is similar or not.

2 Dataset Overview/ EDA

This data set contains around 4 lacs questions pairs with the following features and a label.

- Our data is present in the file named train.csv
- id - represents the id's of the question pairs of the training data set.
- qid1, qid2 - These represents the unique id's of each question which is present in train.csv
- question1, question2 - These are the the full text of each question in a tuple present in train.csv
- is_duplicate - This is our target value, 1 represents that the questions in a tuple have same meaning(duplicates) and 0 if they are not

Out[3]:

	id	qid1	qid2	question1	question2	is_duplicate
0	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	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	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	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	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	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	6	13	14	Should I buy tiago?	What keeps childern active and far from phone and video games?	0
7	7	15	16	How can I be a good geologist?	What should I do to be a great geologist?	1
8	8	17	18	When do you use シ instead of し?	When do you use "&" instead of "and"?	0
9	9	19	20	Motorola (company): Can I hack my Charter Motorolla DCX3400?	How do I hack Motorola DCX3400 for free internet?	0
10	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

Figure 1: Initial Visualisation of Data set

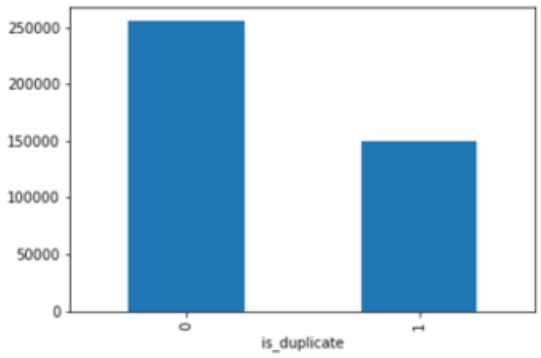

Figure 2: Class-wise distribution of Labels in training data

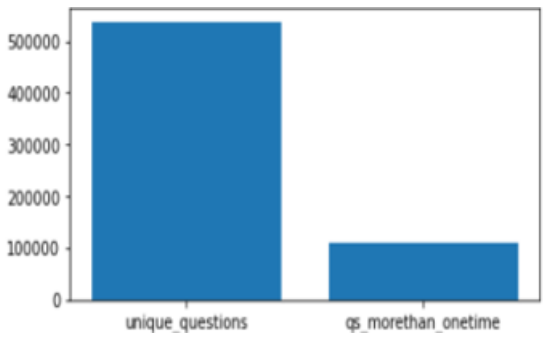

Figure 3: Unique and Repeated Questions

Figure 2 shows that there are around 2.5 lac question pairs which doesn't have same meaning and around 1.5 lac question pairs have same meaning.

Question pairs that are not Similar: 63.1%
Question pairs that are Similar: 36.9%

Figure 3 shows that there are around 5.3 lac questions occurred only on time and around 1.1 lac questions which occurred more than one time.

Figure 4: Word Cloud for Duplicate Question Pairs

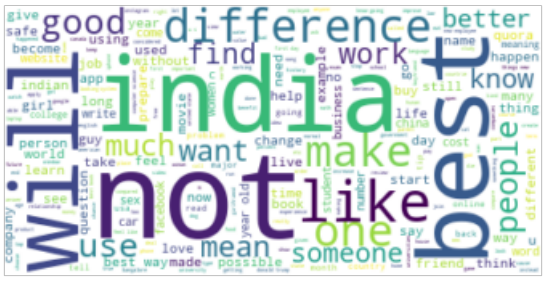

Figure 5: Word Cloud for Non-Duplicate Question Pairs

Word Clouds

3 Related Work

a Recognition Textual Entailment(RTE)**[1]** is a problem in which the relationship between two human written text pairs which are labeled manually for classification with the label entailment are learnt and predicted further.
The difference between labeled entailment and our problem is that we are finding the similarity between the question pairs, our problem classifies certain types of pairs which are questions, whereas in text entailment we are finding the relation between human text pairs, which need to specifically be questions.

b Shubham Shrivastva,**[2]** used NLP's Word2Vec feature embeddings to encode every question pair. This uses pre-trained non contextual word embeddings generated on very large language models using techniques of neural networks. After training, these models can detect similar words which helps in better detection of the pair similarity.
In our problem we are not using Word2Vec because we are focusing on classical ML models.

c Basic feature engineering to encapsulate non-contextual and non-semantic similarities based on commonalities between the question pairs**[3]**, like word share, common words and total words.

4 Methodology

4.1 Problem

Binary Classification problem to predict whether a pair of questions are duplicate or not

4.2 Goal

a The cost of a mis-classification can be very high.(precision of much value than recall).

b You would want a probability of a pair of questions to be duplicates so that you can choose any threshold of choice.

4.3 Feature Extraction

4.3.1 Basic Feature Extraction

Below mentioned are some basic features generated before preprocessing :-

- freq_qid1 = Count of first question of a tuple occurred on whole dataset(including question 1 and question 2 column).
- freq_qid2 = Count of second question of a tuple occurred on whole dataset(including question 1 and question 2 column).
- q1len = Length of the first question of the tuple in dataset.
- q2len = Length of the second question of the tuple in dataset.
- q1_n_words = Count of total words present in first question in a tuple of the dataset.
- q2_n_words = Count of total words present in second question in a tuple of the dataset.
- word_Common = Count of common words(unique) present in question 1 and question 2 of a tuple in the dataset.
- word_Total =Count of total words present in first and second question in a tuple in the dataset.
- word_share = ratio of the common words to the total words.
- freq_q1+freq_q2 = total of frequency of question 1 and question 2 in a tuple of the dataset.
- freq_q1-freq_q2 = absolute difference of frequency of question 1 and question 2 in a tuple of the dataset.

4.3.2 Advanced feature Extraction

Firstly, we did the following preprocessing on the given dataset.

a Removing html tags

b Removing Punctuations

c Performing stemming

d Removing Stopwords,etc.

After this, we added the following features further on the existing dataframe. The following definitions are used in the explanations of features ahead :-

- Token: refers to space-separated items of every sentence.
- Stopwords: commonly occurring lesser relevant words as defined in nltk.

- Word: subset of tokens which are not stopwords.
- cwc_min : Ratio of count of common words to minimum number of words in question 1 and question 2.
- cwc_max : Ratio of count of common words to maximum number of word in question 1 and question 2.
- csc_min : Ratio of count of common stop words to minimum number of stop word in question 1 and question 2.
- csc_max : Ratio of count of common stop words to maximum number of stop word in question 1 and question 2.
- ctc_min : Ratio of count of common token words to minimum number of token word in question 1 and question 2
- ctc_max : Ratio of count of common token words to maximum number of token word in question 1 and question 2
- last_word_eq : Checking if last word of both two questions is equal or not
- first_word_eq : Check if First word of both two questions is equal or not.
- abs_len_diff : Absolute number of words difference.
- mean_len : Average Token Length of both Questions
- fuzz_ratio : It is the ratio of edit distance of Q1 to Q2.
- fuzz_partial_ratio : It performs fuzz_ratio on substrings. Takes the shorter Q and compares it with all possible substrings that can be generated from Bigger Q.
- token_sort_ratio : First all the punctuations are removed,all letters are converted to lowercase and strings get converted into tokens. They are joined after alphabetic sorting. Then finally the fuzz ratio is calculated between these strings.
- token_set_ratio : First take intersecting words between strings then tokenize and sort them. Put them together. At last calculate the token sort ratio on them. Repeated or same words are of no use.
- longest_substr_ratio : Ratio of length longest common substring to min length of token count of Q1 and Q2

4.4 Feature Analysis and Visualization

Below are the analysis for relevance of some features generated using feature engineering.

4.4.1 Word Share

From the above plots we can check the plot of non-duplicate pair and the duplicate value is overlapping a bit.

From the above violin plot and distribution plot, we observed that when the questions are duplicate, the word share ratio is more as compared to when they are not.

From that, our word share feature is helpful to some extent in decision-making whether our question pair is similar or not.

The curve is overlapping, so we cannot decisively say whether our question pair is similar. If there would have been no overlapping, then the word share feature would solve our entire problem, and that feature would be our decision feature for classifying duplicacy.

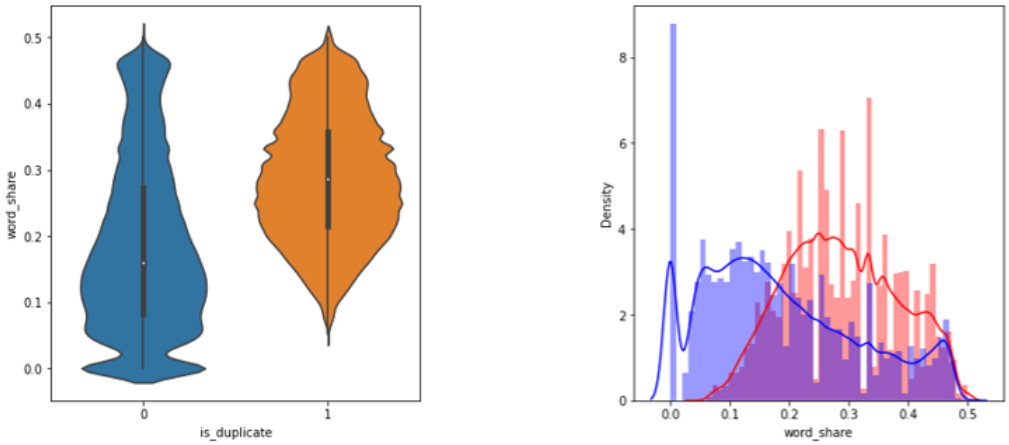

Figure 6: Violin Plot and Distribution plot of Is_Duplicate and Word Share

For example, if the curve for the non-similar question would have ended at a 70% word_share and the curve for the similar questions would have started from a 70% word_share, then word share would be deciding feature such that if the word share percentage is more than 70 then declare question pair is similar otherwise not.

4.4.2 Word Common

Plot for this curve is almost overlapping.

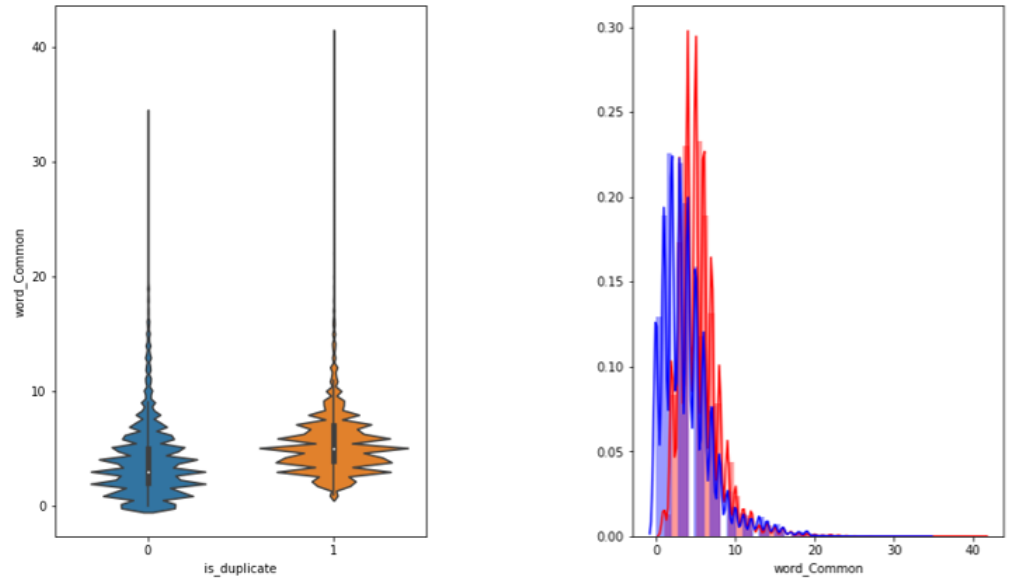

Figure 7: Violin Plot and Distribution plot of Is_Duplicate and Word Common

From the above violin plot and distribution plot, we can see that word_common feature is much more overlapping and hence would be of very little help in decision-making.

4.4.3 FuzzyWuzzy Features

Using the fuzzywuzzy library, one of the simplest methods, we may have a score out of 100 that indicates the equality of two strings by providing a similarity index.

Python's FuzzyWuzzy module is used for string matching. Finding strings that match a certain pattern is known as fuzzy string matching. In essence, it calculates the differences between sequences using Edit Distance.

Figure 8: Scatter Plot Between All FuzzyWuzzy Features

Figure Above is showing how related Different FuzzyWuzzy Features are with each other. We can see some amount of overlapping but not completely. This means every feature has some importance which finding whether the questions are duplicate or not.

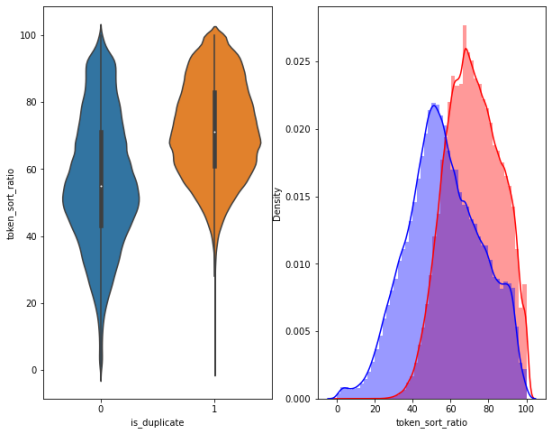

Figure 9: Violin Plot and Distribution plot of Is_Duplicate and token_sort_ratio

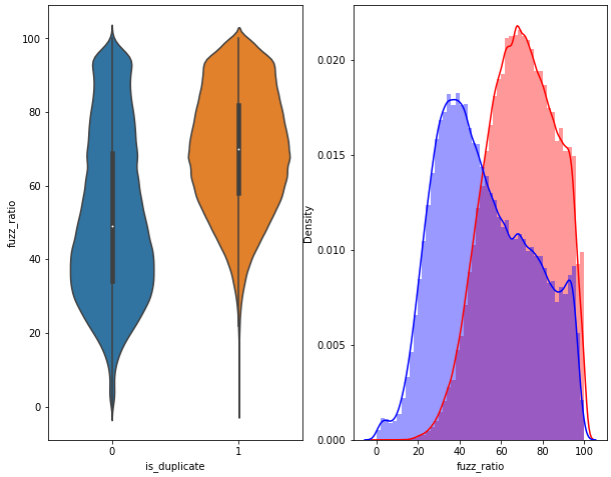

Figure 10: Violin Plot and Distribution plot of Is_Duplicate and fuzz_ratio

4.5 Models

We have used several models in our project. Lets bring some light to what these models are :-

4.5.1 Logistic Regression

In logistic regression, the model estimates the probability of occurring of the event such as true or false on the basis of the dataset of independent variables or features.
The output is probabilistic, so the dependent variable is bounded between 0 and 1.

$$h_\Theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{1}$$

4.5.2 Linear SVM

Linear SVM algorithm is a maximum margin classifier that creates a linear decision boundary between data points of the two classes such that there is maximum separation between the closest set of points from opposite classes. The decision boundary is called the separating hyperplane, and the nearest points to this boundary are called the support vectors.The algorithm's goal is to find a hyperplane that maximizes the distance between these support vectors.

$$J(\Theta) = \sum_{i=1}^{m} y^{(i)} max(0, 1 - \theta^T x) + (1 - y^{(i)}) max(0, 1 + \theta^T x) \tag{2}$$

4.5.3 Random Forest

Random forest is the supervised machine learning model which uses different number of decision tree with subset of features and/or sampels and take the suitable vote which will help in classification problem, the output will be the value which is selected by most of the decision trees for classification(class) and mean for regression(integer value). Training set X = $x_1, ..., x_n$ with responses Y = $y_1, ..., y_n$, bagging repeatedly (B times)

$$\hat{f} = \frac{1}{B} * \sum_{b=1} B f_b(x') \tag{3}$$

Decision tree entropy:

$$Entropy(node) = -\sum_{i=1}^{c} p_k log(p_k) \tag{4}$$

$$p_k = \frac{number\ of\ observation\ of\ class\ k}{all\ observations\ in\ node}$$

4.5.4 XGBoost

XG Boost is the ensemble machine learning algorithm based on a decision tree which uses a gradient boosting framework.Gradient boosting is an approach in which it gives a prediction model in the form of a group of weak prediction models. It generally performs better than random forest because it allows optimisation on boosting.

5 Experimental Results

5.1 Performance Metric

5.1.1 Log-Loss(Best Metric)

Log loss returns the probability of predicted labels to the labels of its training data. Logistic regression and its enhancements uses log loss as its metric. For a logistic model it may be defined as its negative log likelihood. Log Loss of an instance is calculated using formula give below:-

$$Logloss_i = -(y_i * log(p_i) + (1 - y_i) * log(1 - p_i)) \tag{5}$$

where,
i: a given instance
y: actual value
p: prediction probability

Overall Log Loss will be given by:-

$$Logloss = \frac{1}{N} * \sum_{i=1}^{N} Logloss_i \tag{6}$$

Reason for log loss being best metric
We shall choose log loss as our best metric as in log loss, we predict the probability of a certain class label for every sample.On the other hand, accuracy predicts the binary value for the sample belonging to a particular class. In our problem, log loss is a superior metric because our main motive is to get a probabilistic value of class label prediction rather than a straight value, so we could have the freedom to predict the similarity based on a threshold. Setting this threshold is important because the business cost of misclassification is very high. For example, if the question pair is duplicate and we predicted it to be non-duplicated then is a smaller problem but if the question pair is non-duplicated and we predicted it duplicated then that maps to a very high business cost. In our project we are setting the threshold value to 50 percent using np.mean function

5.1.2 Precision

Precision is a measure we use when our aim is to minimize the false positives. As, high precision means low false positives and when false positives are high then precision becomes low. It doesn't care about true negatives and false negatives. The formula for Precision is:-

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{7}$$

5.1.3 Recall

Recall is a measure we use when our aim is to minimize the false negatives. As, high recall means low false negatives and when false negatives are high then recall becomes low. It doesn't care about false positives and true negatives. The formula for Recall is:-

$$Precision = \frac{TruePositive}{TruePositive + FalseNegatives} \tag{8}$$

5.1.4 Accuracy

Accuracy is the most commonly used metric. It is just the ratio between correct predictions and total predictions. It is used when your aim is just to maximize the correct predictions. The formula

for Accuracy is:-

$$Precision = \frac{TruePositive + TrueNeagtives}{TruePositive + TrueNegatives + FalsePositives + FalseNegatives} \quad (9)$$

5.2 Logistic Regression

The first thing we did was to run Logistic Regression with all our preprocessed datasets. Because it is very powerful and efficient when the dimensions of the data are higher. We are choosing logistic regression We have used a log loss function and l2 regulariser . We have train to on different alpha like $10^-5, 10^-4$,..... 1, 10 to fine tune parameters for the model. We will plot the log loss against the alpha values so that we can further choose a range of alpha to further fine tune the hyperparameter to get minimum log loss. And choose the best param with minimum log loss. We got 76% precision , 65% recall , 79% accuracy and log loss error is 40.89%.

5.3 Linear SVM

Now we are using linear SVM. We are training a model with alpha like $10^-5, 10^-4$,..... 1, 10 to fine tune parameters for the model. We will plot the log loss against the alpha values so that we can further choose a range of alpha to further fine tune the hyperparameter to get minimum log loss. And choose the best param with minimum log loss. We got 76% precision , 65% recall , 79% accuracy and log loss error is 40.40%. Important thing to note here is that why we are using log loss, here precision , recall and accuracy is the same, linear SVM model able to better predict the target class with a good probability. Hence we are using the log loss to compare our models.

5.4 Random Forest

We are using random forest to try to reduce log loss or improve precision and recall, w.r.t to linear SVM and logistic regression. We are training on different estimators and max depth of all the decision trees in the forest to fine tune the hyper parameter and after that we are choosing the best parameter which has the lowest log loss on test data. The log loss error is 39.77% on the test data.

5.5 XgBoost

We are using Xgboost to try to reduce log loss or improve precision and recall, w.r.t to linear SVM, logistic regression and random forest . We are training on different estimators and max depth of all the decision trees in the forest to fine tune the hyper parameter and after that we are choosing the best parameter which has the lowest log loss on test data.The log loss error is 37.25% on the test data

5.6 Summary

Table 1: Summary of Results

Type of Model	Features Used	Log Loss	Precision	Recall	Accuracy
Random Model	Random Data	0.8829	0.6547	0.6967	0.7779
Logistic Regression	Basic Features + Advance Features + TF-IDF	0.4089	0.76	0.65	0.79
Linear SVM	Basic Features + Advance Features + TF-IDF	0.4040	0.76	0.65	0.79
Random Forest	Basic Features + Advance Features + TF-IDF	0.3977	0.78	0.69	0.81
XGBoost	Basic Features + Advance Features + TF-IDF	0.3725	0.78	0.70	0.82