Quora Question Pairs

Sina Darban Khales

Introduction

What is the problem?

Quora is a place to gain and share knowledge—about anything. It's a platform to ask questions and connect with people who contribute unique insights and quality answers. This empowers people to learn from each other and to better understand the world.

Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded questions. Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question. Quora values canonical questions because they provide a better experience to active seekers and writers and offer more value to both of these groups in the long term.

Identify which questions asked on Quora are duplicates of questions that have already been asked. This could be useful to instantly provide answers to questions that have already been answered. We are tasked with predicting whether a pair of questions are duplicates or not.

Data sources:

Kaggle data was used for the analysis. The dataset comprised 404,290 pairs of questions (question1 and question2), with a binary column serving as the ground truth, indicating 0 for non-duplicated pairs and 1 for duplicated question pairs.

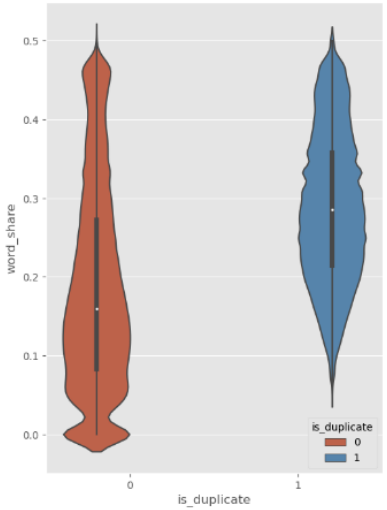
Data Wrangling

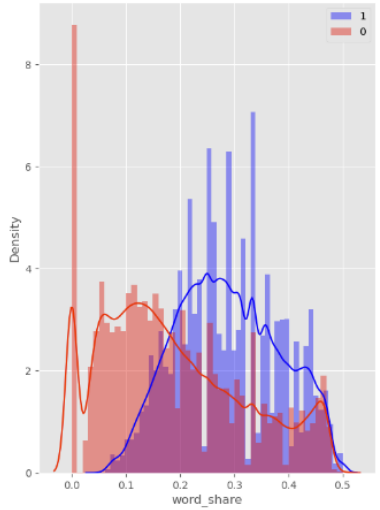
I first examined at the question pairs and then individual questions. Looking at the data, 39.25% of question pairs were similar, while none of them are identical. Also, there were 537,361 unique questions in our dataset with 271,219 questions appearing more than once. That was about 33.54% of all questions.

The following table shows the five most frequently asked questions.

	question	n
0	What are the best ways to lose weight?	161
1	How can you look at someone's private Instagra...	120
2	How can I lose weight quickly?	111
3	What's the easiest way to make money online?	88
4	Can you see who views your Instagram?	79

In the first step, I found that 3 records had missing either question 1 or question 2. Those records were removed from the analysis. Next, I extracted some basic features from the individual questions and question pairs. The basic features were: The frequency of question1 and question2 for each row, the length of the questions for each row, the number of words in each question, the number of common words in question1 and question2, the total number of words in both questions, the number of common words in question pairs/the total number of words in question1 and question2, the total number of words in question pairs, and the absolute length difference between questions in each row.
Looking at the plots for the number of common words in question pairs/the total number of words in question1 and question2 (word_share), it was evident that the median was greater when questions were similar.





Preprocessing of text:

This step included lowercasing the questions, removing the html tags, and some manual replacements. After that I tokenized the questions and separate them based on being either an English stop word or a none-stop word.

Advance feature extraction:

After preprocessing, I conducted advanced feature extraction. This involved identifying common words, common stop words, and common non-stop words, and dividing them based on the minimum and maximum length of the questions in each question pair. I also verified whether the first and last words of the questions in each pair were identical. Additionally, I computed the average token length and token length difference for each question pair. Fuzzy matching techniques, such as partial fuzzy ratio, token set ratio, and token sort ratio, were also applied to measure similarity between each question pair. Finally, Levenshtein distance, word embedding similarity, cosine similarity, and cosine similarity based on TF-IDF vectors were calculated to quantify the similarity of questions in each question pair.

Exploratory Data Analysis:

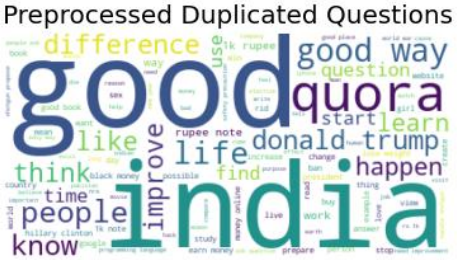
To gain insight into the most frequently occurring words in both duplicated and non-duplicated question pairs, I generated word clouds. The findings revealed significant similarity in the vocabulary utilized within both duplicated and non-duplicated question pairs.

Consequently, I recognized the necessity of analyzing question pairs on an individual basis rather than treating duplicated and non-duplicated batches as distinct entities.

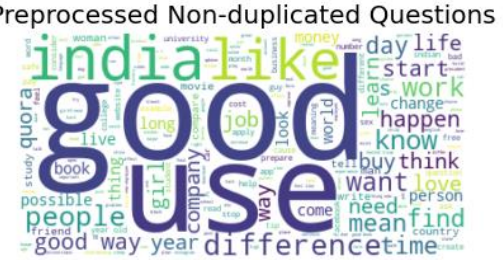
Original Duplicated Questions

Original Non-Duplicated Questions

Preprocessed Duplicated Questions



Preprocessed Non-duplicated Questions



Data Preprocessing:

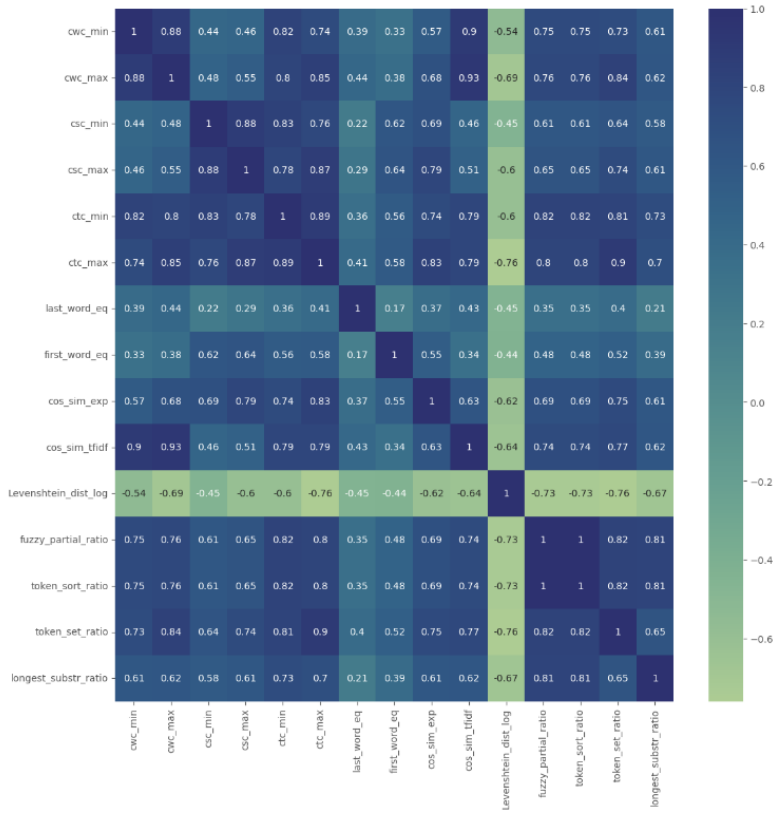
This stage involved examining the data distribution, transforming the data, identifying intercorrelations, and normalizing the data. I observed that the Levenshtein distance was right skewness, requiring the utilization of a log-transformation. Conversely, cosine similarity displayed left skewness, requiring an exponential transformation for normalization.

After normalizing the data using min-max normalization, the next step was to find the important features. For that, I used random forest feature extraction first. Next, I created the correlation matrix to find the linear correlation between the independent variables. After that, I used multiple methods such as variance threshold, recursive feature elimination, and Lasso regression to find the best set of features for modeling. The features that I picked for the modeling part were

'cos_sim_exp' 'cos_sim_tfidf', 'cwc_max', 'token_set_ratio', 'Levenshtein_dist_log', 'longest_substr_ratio', 'csc_max', 'last_word_eq', 'first_word_eq']

FEATURE SELECTION: 1. RANDOM FOREST

Feature	Importance
cos_sim_exp	0.10484
cwc_max	0.092286
cos_sim_tfidf	0.092234
token_set_ratio	0.087169
ctc_max	0.080445
Levenshtein_dist_log	0.077962
fuzzy_partial_ratio	0.076135
cwc_min	0.07587
token_sort_ratio	0.068369
longest_substr_ratio	0.06354
ctc_min	0.053605
csc_max	0.050648
csc_min	0.040017
last_word_eq	0.02777
first_word_eq	0.00911



Model Development:

For the machine learning modeling part, I first split the data into a training set and a test set, trained the models using the training set, and then find the performance metrics for each model, using the unseen data or the test set. The ML models that I chose were:

- Logistic regression
- Random Forest
- XGBoost
- Gaussian Naïve Bayes

All models were tuned using the proper hyperparameters and the best model was used to calculate the predictive performance.

As a second approach I trained similar models, but this time used the tfidf vectors only, instead of the features that I extracted in the previous steps. The model results were as follows:

Model	Accuracy	AUC	Recall	Precision
Logistic regression	0.69	0.77	0.47	0.60
Random forest	0.76	0.85	0.70	0.67
XGBoost	0.76	0.84	0.68	0.69
Gaussian Naïve Bayes	0.68	0.75	0.66	0.75
Logistic regression-tfidf	0.75	0.80	0.53	0.72
Random forest-tfidf	0.81	0.87	0.70	0.87
XGBoost-tfidf	0.74	0.79	0.40	0.77

Based on the performance metrics, the random forest model based on the TF-IDF vectors were found to outperform all the other investigated models.

Conclusion

- Using the proposed models, the website can label the question pairs as either duplicated or non-duplicated
- The models help with providing answers instantly to questions that have already been answered
- This will help the responders to build on the previous responses and avoid duplicated answers to the same questions.
- Building models based on transformers such as BERT might lead to better results.