Data Analytics
Università della Svizzera italiana

Project 1 - Quora Duplicate Questions

Bhardwaj Lovnesh

April 15, 2024

Please note that this project was conducted alone as I was not able to contact the person that I was supposed to do it with. This has already been alerted to the TAs in one of the lectures.

Introduction

Quora is a popular website where people ask questions and express their concerns. However, since there are many users of the said website, undoubtedly, some of the questions that are already asked by a person is asked reapeatedly due to multiple factors, including but not limited to users neglecting to research existing queries before posting their own, etc. This creates a position of difficulty for the people who are maintaining the website, and also the people who are searching for the answer to a question who might get frustrated amidst the redundancy. Due to this, Quora employs a multitude of mechanisms such as the use of algorithms to detect similar questions, the use of moderators in the space.

Algorithms play a key role in identifying the similar questions due to a multitude of factors but above all, their efficiency. However, algorithms are not infallible, there may be the case that there are false negatives/positives, which is a situation that we want to avoid. To somewhat mitigate this, people use the means of supervised learning approaches to teach the system to mimic a human in segregating the duplicate questions from the non-duplicate questions. Therefore, we will be taking a similar approach to find the duplicates. Our approaches, which are two, will be based on training a simple transformer as seen in the lab lectures to see how we perform and if we are able to do well. Our second approach will be based on pre-processing the data, implementing concepts such as 'Jaccard Similarity' of sentences, and some methods such as stop-word removal, stemming, followed by TF-IDF to add some syntactic meaning to the sentences, and then training a Logistic Regression module. The rest of the report will be structured as follows: Section 'Exploratory Data Analysis' focuses on plotting some basic plots and histograms in order to better understand the data that we are working with. Section 'Data Pre-processing and Results' deals with the data that we have at hand, and pre-processes it to make the data that we are going to be using at a later stage all the while talking about the results of fitting. Finally, the last section 'Conclusion and Discussion' deals with explaining the results, drawing conclusions from it, and discussing about the points to improve with respect to the designing and approach to the problem.

Exploratory Data Analysis

This section will explore the data that we are given. Please note that the data used for exploration is found in the file **train.csv**, which contains the training data. Reading this file as a panadas dataframe shows us that the dataframe includes the following columns: *id, qid1, qid2, question1, question2, is_duplicate*. The columns of importance are the last three, which contain questions 1 and 2, and if they are duplicates or not, represented by 0 or 1, respectively. This alone does not give us much information regarding the similarity of the questions. Of course, from a human perspective, one can go through all of them and say if they are duplicates of each other. However, doing so would be inefficient and would not scale well, which we have touched upon briefly above. First, I would like to understand how the duplicates are distributed.

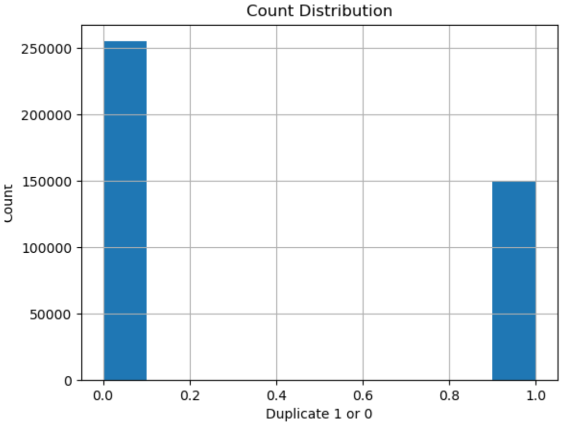

Figure 1: Duplicate vs Non-Duplicate Count

We can see from Figure (1) that the data is not divided equally amongst either of the classes. This might create some bias later when we train a classifier towards the non-duplicate variable. However, one can say that the bias induced would not be too much as there is a lot of data between both classes. Afterwards, I added other columns to the dataframe to better understand how the data is distributed. I calculated the length of each question and put them into new columns: *len_q1, and len_q2*. Now, I would like to see how the lengths of each of the questions are distributed.

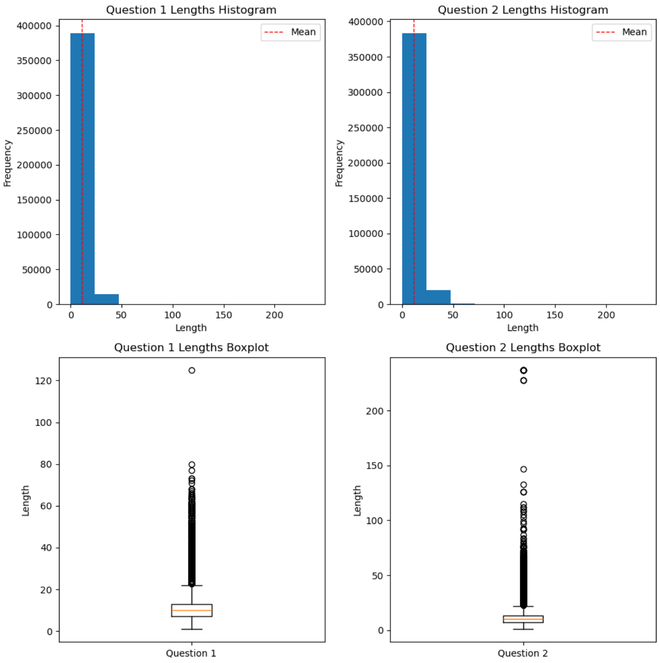

Figure 2: Question Length Comparison

From Figure (2), we can see that the distribution is distributed about the same. However, there is a difference in the length of the two questions, whereby the length of the second question has a lot of outliers. However, we can attribute it to the data collection process, which might not induce any bias in the training of the model. The next step would be to see how the distribution of the difference in the length between both the questions is in the case of duplicates and non-duplicates.

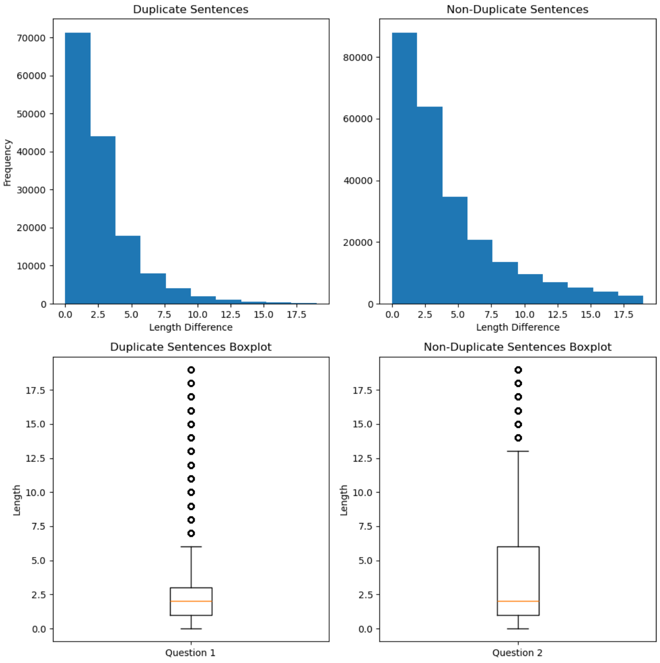

Figure 3: Length Difference Comparison - Duplicates vs. Non-Duplicates

Now, we have something that might have a pronounced impact on the model's prediction quality. Please note that this is a focused version of the plot (up to sentence length 20); another plot shows the complete difference in the distribution, which I do not include here as the report would be too cluttered. We can see from Figure (3) that the question length difference distribution between the duplicate sentences and the non-duplicates is lesser in the duplicates. The distribution of the duplicates here is also more focused on the mean and not spreading out, which is not the case regarding the non-duplicates. This, therefore, can be used as a feature for our model.

Data Pre-processing and Results

When it comes to pre-processing the data for our model, there were multiple strategies that I could have undertaken. I will first describe the strategy used for the Logistic Regression model. To provide meaningful features to the model and for the resulting prediction to be accurate, I combined a multitude of concepts that are used in Natural Language Processing tasks. First, I took both the questions and pre-processed them in a way that would remove the stop words from them and stem the words as well. Doing so would give us a better representation of the meaningful words in the sentence, all the while removing the non-necessary words such as 'a', 'the', 'to', etc. and also stem them to their base form for better comparison between the pairs, an example of the same can be that 'pairs', and 'paired' stem to the same word 'pair'. This is done because different sentences might use the word in a certain format but the semantics of which might be similar, which is pivotal for comparisons. After this, I also used the TF-IDF to capture the semantic information through the weighting scheme of the TF-IDF method, which has a higher weight associated with rarer terms. Further, on the stemmed data, I calculated the Jaccard Similarity simply by using the words as the elements of the set since the Jaccard similarity measure is useful in capturing the semantic structure through the word level parsing and how close the two questions are. Jaccard Similarity measure However, it will be biased towards sentences that have similar wordings and the same syntactic meaning. This means that even though the sentences "The cat is on a mat" and "The mat has a cat" have similar meanings, they would record low Jaccard Similarity. All these methods were recorded as a column in the dataframe Later, the dataframe was split into a test and a training dataset. The features for the training dataset were the difference in the length of the questions, the TF-IDF measure, and the Jaccard Similarity measure, while the target vector is a categorical 0 or 1 vector given by the 'is_duplicates' column. All this is then fed to the Logistic Regression classifier, producing a model with about 78%

The second way I tried was by fine-tuning a pre-trained Bert classifier, as shown in the lab lectures, which yielded an accuracy of about 83% on the test set. There was no pre-processing of significance done except for the fact that the questions in the dataset were tokenised to a maximum length of 20 as they were the ones till most of the distribution lay (as inferred from Figure (2)). After this, the sentences were tokenised using a pre-trained tokeniser and fed to the model for one epoch.

Judging from a human eye, the logistic regression testing done on the test set provides us with a good enough text classification. For example, the sentence, "Can a vacuum cleaner concentrate suck your eye out if it is pressed against your face?" and "Could a vacuum cleaner suck get your eye out if directly pressed on the face?" were judged to be similar, which is what we expect. Another example of the sentences that were judged to be correct is, “Why do people ask find on Quora that could simply be googled?” and “Why do people ask questions everyone Quora they could easily search via Google, Bing or Wikipedia?”. However, some sentences which we expect to be non-duplicates were classified as duplicates as well, an example is “What site the best example of dedication in any field?” and “What are some of the best examples of new in any field?”. This can be attributed to the many words shared between the sentences and the low weight that the word ‘new’ might have in the TF-IDF vector.

Conclusion and Discussion

To summarise the findings and the study results, we can say that the classifiers performed well, primarily the hand-trained ones. There were some false positives, but they can be mitigated using better techniques such as POS tagging, dependency parsing, or syntactic tree representations. Doing so would increase the number of features in the design matrix, and potentially increase the quality of the model.

The type of model used is also crucial. One can increase the complexity of the model, the Logistic Regression model in this case, to better encapsulate the syntactic similarities between the sentences. However, when it comes to raw classification with only a few features, the result that we got is a good one.

However, a case can be made on how the data is collected and if it only contains English words. For example, there was a question that asked about the difference between two Japanese letters that are pronounced the same. However, this character encoding could not be captured by our model, and thus resulted in a biased model. One can better the model by adding numerous layers of complexity.