Quora Duplicates Question Detection

Rishab Ghanti

rishabrghanti@gmail.com

Contents:

1 Definition:

1.1 Project Overview

Quora is a very famous platform (website and Android app) all over the world where questions can asked by anyone in the community which are answered by Quora's 100 million monthly active users. There are different platforms like *StackOverFlow*[1], *Yahoo! Answers*[2], *WikiAnswers*[3] etc. for different domains but Quora[4] is by far the most popular general question-answer platform.
Due to its huge user base, it is expected that many users ask questions structured differently but having the same meaning. For example, the questions 'What is the population of USA' and 'how many people live in USA' are duplicates. Marking both these as duplicates and showing users one version of the question with the correct answer will allow users to get answers to their questions quickly and giving them more value.

1.2 Problem Statement

The goal of this project is to identify if a question pair is duplicate, i.e. the question pair has the same meaning. This is a natural language processing problem which is a popular field in Artificial Intelligence.

For example, we have a question pair as follows:

- How much does a Tesla model S cost?
- What is the cost of a Tesla model S?

We know that these 2 questions mean the same thing and will have the same answer. Hence the system will flag this pair as a duplicate.

To build such a mechanism in place I did the following steps:

1. Downloaded the Quora dataset and did some pre-processing on it to get a better understanding of it.
2. Split the dataset into 90% as training set and 10% as testing set.
3. Further split the training set into 10% as validation set and used the remaining data to train the model.
4. Extracted usable features from the sentences to feed to the model.
5. Using the training data trained a few models with different parameters to be able to recognize duplicate and non-duplicate pairs.
6. Tested these models and picked the one with the best score.

1.3 Metrics:

After exploring the data set I see that the dataset has 36.91% of positive entries whereas it has 63.08% of negative entries since the dataset is unbalanced, the metric that I plan to use and will best suit this application to determine the accuracy of a model is F-score[5]:

$$F = 2 * \left(\frac{precision * recall}{precision + recall}\right)$$

precision – number of positive results divided by the number of positive results returned by the classifier

recall – number of correct positive results divided by the number of all relevant samples (all relevant samples that should have been identified as positive)

2 Analysis

2.1 Data Exploration

Below is the head of the Quora dataset

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0
5	5	11	12	Astrology: I am a Capricorn Sun Cap moon and c...	I'm a triple Capricorn (Sun, Moon and ascendan...	1
6	6	13	14	Should I buy tiago?	What keeps childern active and far from phone ...	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 Moto...	How do I hack Motorola DCX3400 for free internet?	0
10	10	21	22	Method to find separation of slits using fresn...	What are some of the things technicians can te...	0
11	11	23	24	How do I read and find my YouTube comments?	How can I see all my Youtube comments?	1
12	12	25	26	What can make Physics easy to learn?	How can you make physics easy to learn?	1
13	13	27	28	What was your first sexual experience like?	What was your first sexual experience?	1

These are the column definitions:

Id – ID to identify the question pair

Qid1 – ID of the first question pair

Qid2 – ID of the second question pair

Question 1 – Text of the first question in the question pair

Question 2 – Text of the second question in the question pair

Is_duplicate – Quora reviewers decision if the questions pair is duplicate or not.

2.1.1 Summary Statistics

1. Total number of entries – 4,04,290

2. Total number of positive entries – 1,49,263

3. Total number of negative entries – 2,55,027

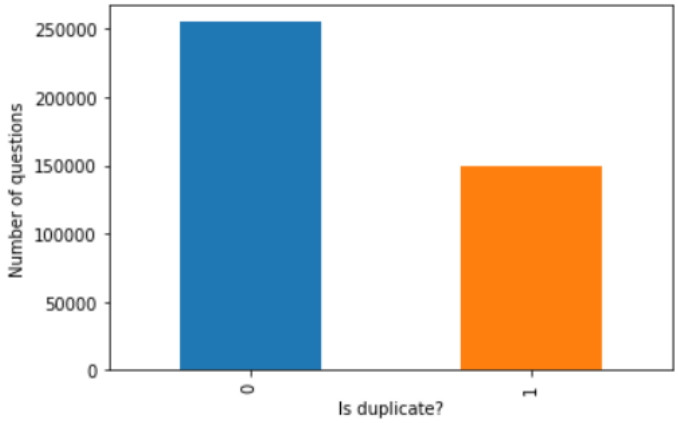
4. Percent positive entries – 36.91%

5. Percent negative entries – 63.08%

2.2 Data Visualization:

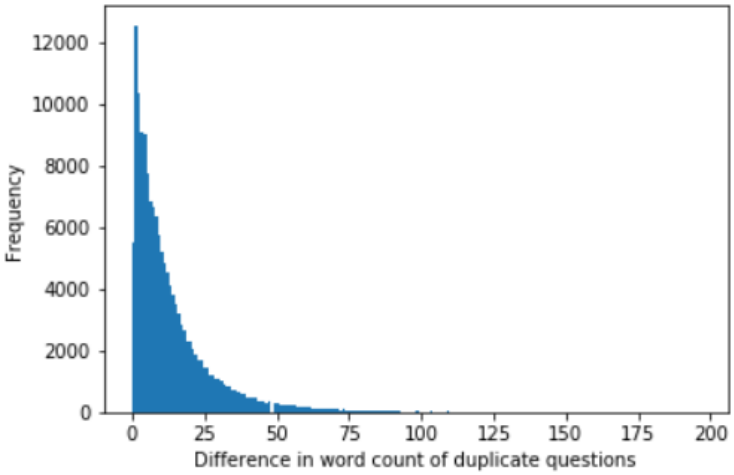
2.2.1 Class Distribution

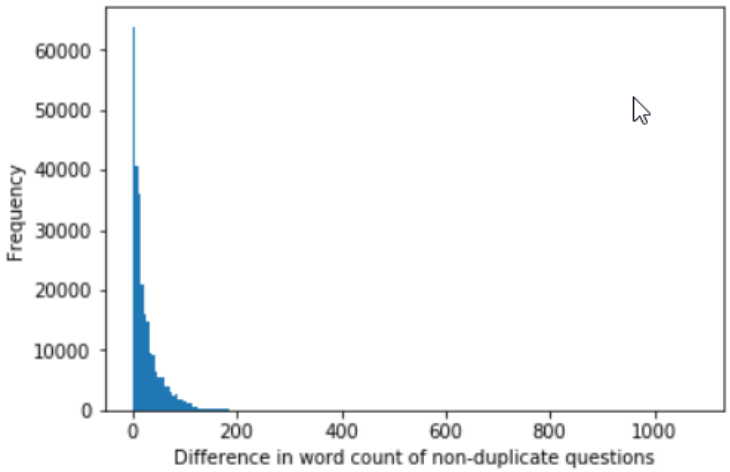
Below is a graph explaining the class distribution of duplicates and non-duplicate pairs.



2.2.2 Word Count Difference

In this section we will discuss how the word count difference influences if a question is a duplicate or not.





Observation from the above figures:

1. For duplicates, the word count tapers smoothly.
2. For non-duplicates, there is a sharp decrease In the 2nd bin.
3. There are question pairs that have word count difference 75 and above and are still duplicates, hence I conclude that a simple bag of words model will not work for this dataset.

2.2.3 Jaccard index

I have considered the bag of words model as a baseline model. The Jaccard index formula is:

$$J(A,B) = \frac{No.of\ common\ words\ in\ question1\ and\ question2}{No.of\ unique\ words\ in\ question1\ andquestion2}$$

where A and B are words in the questions and hence the equation can also be written as

$$J(A,B) = \frac{A \cap B}{A \cup B}$$

Using the above formula I concluded the following:

1. For duplicates, Jaccard similarity is like a gamma distribution. Jaccard is bad at predicting if a question pair is duplicate or not.
2. For non-duplicates, Jaccard similarity is again a gamma distribution but. Jaccard is bad at predicting if a question pair is duplicate or not.

2.3 Algorithms and Techniques

2.3.1 Feature Extraction

The inputs to the algorithm are 2 strings of variable length. After the input has been word-tokenized (split into individual words) there are primarily two ways in NLP (Natural Language Processing) using which features can be extracted. As explained below.

Bag of words

Here, every word is assigned a unique number. Hence the input now is a one-hot encoded array of numbers representing those words.

Consider a text corpus having the words ‘today’ and ‘tomorrow’. The word ‘today’ will be assigned the number 0 while ‘tomorrow’ will be assigned 1. Hence in one-hot encoded representation the word ‘today’ would be represented as [1 0] and ‘tomorrow’ would be [0 1].

Hence the number of features extracted would equal the total number of unique words in the corpus (training set).

Word Vectors

The disadvantage of using bag of words technique is that we cannot use it for large corpora datasets since it will not scale. This problem is solved by using word vectors since it creates fixed length vectors instead of sparse arrays.

As opposed to bag of words, word vectors have to be trained on a corpus before they can be used to extract features from different corpora. Thankfully there are a lot of pre-trained models available for us to use.

Two popular algorithms to generate them are *Word2Vec*[6] and GloVe[7]. SpaCy[8] comes with a GloVe model of 300 dimensional vectors trained on the common crawl corpus.

I will be using the word vectors approach since it will allow me to train for words that do not appear in the Quora training corpus but a similar word does.

2.3.2 Pooling

The pair of questions in the dataset may be of different sizes each. The word vector of the questions would be a matrix of size m*300 where 300 is the expected word vector size and m is the number of words in the question.
For a pair, m could be different since each question could have different number of words. Hence we need to normalize the matrix. There are 2 ways to do this:

1. Max pooling: Take the maximum of each column in the m*300 matrix to get a 1*300 matrix.
2. Mean pooling: Take the mean of each column in the m*300 matrix to get a 1*300 matrix.

Max pooling is reported to perform better[9]. To get better accuracy I have concatenated mean and max pooling to get a matrix 1*600 in dimension. Thus, every question is converted into a 600-dimensioal vector.

2.3.3 Neural Networks:

For this project I have used a deep neural network to create the model. Neural networks have started being used more frequently to solve NLP problems.

Neural networks learn to model a problem similar to supervised learning but the deeper (more number of layers) the network the more complex characteristics it will be able to detect. A downside of neural networks is that it is very difficult to correctly design them. The structure itself has many possibilities and we also have many hyper parameters that we can tweak to better the accuracy of the model. These models take a longer time to train as well.

2.3.4 Improving Neural Network Accuracy

These are some of the different techniques to improve the accuracy and reduce overfitting of neural networks

Dropout: In this technique random units in the network are switched in every epoch (step) to force the network to learn redundant representations of the input. Dropout is done only while training the network which helps in better training it. Since the quantity of training data in my case is less, I have not used the dropout technique.

Batch Normalization: A batch normalization layer shifts the inputs from the previous layer to have zero mean and unit variance. This prevents the data flowing into the network to not become very big or small. It is said to result in higher accuracy and faster convergence/learning. I have implemented this in my project with batch normalization on and off.

2.4 Benchmark

There is a well-known problem in the information retrieval field called *near duplicate detection*. One way to solve this is Jaccard[10] similarity bag of is one way of finding near duplicates. This baseline model is run on the 10% test dataset.

3 Methodology

3.1 Data Preprocessing

The Quora training set train.csv is stored in the same directory as the Jupyter Notebook[11] as can be seen in my GitHub repository linked in this report.

3.1.1 Feature Selection

Out of all the data in train.csv we will use only 2 of the columns

question1 – text of first question in the pair

question2 – text of second question in the pair

The target variable is:

is_duplicate – reviewers decision about the question pair if it is a duplicate or not.

3.1.2 Feature Transformation

Every question in the dataset would go through this process. Consider ‘how is the weather’ as a question in the dataset.

Step 1 – Tokenize text into words.

Using spacy the questions were converted into different words (tokenized). For example the above question would be tokenized into ‘how’ ‘is’ ‘the’ ‘weather’.

Step 2 – Get GloVe vector for each word

For each of the word a GloVe vector was generated of length 300. Thus a matrix of size number of token * 300 was generated. In our example the matrix will be of size 4 * 300 as shown below.

how	0	0	…	0
is	0	0	…	0
the	0	0	…	0
weather	0	0	…	0

Step 3 – Mean pooling

The n * 300 matrix was is now converted to 1 * 300 matrix by taking the mean of all the columns.

Step 4 – Max Pooling

Same as ablove a 1 * 300 matrix is generated by taking the max of all the columns.

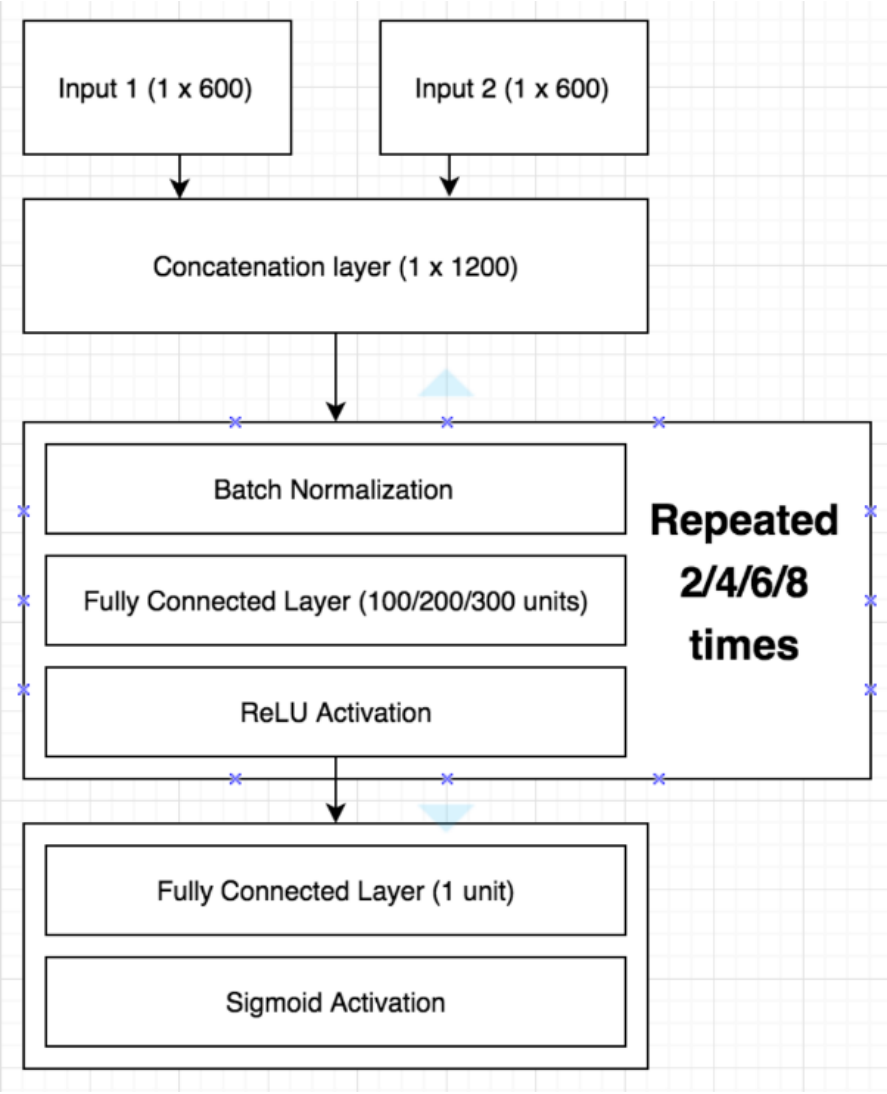
Step 5 – Concatenate mean and max

In this step the mean and max generated above are converted into a matrix of size 1 * 600.

The above steps are repeated for all the questions in the dataset giving us 2 vectors of size 1 * 600 as inputs.

3.2 Implementation and Refinement

3.2.1 Structure of the Neural Network



The following are the features of the model.

1. The input is provided with 2 inputs of size 1*600 of features of the question pairs.
2. These inputs are concatenated into one layer of size 1*1200 in the concatenation layer as shown above.
3. A learning unit consists of a dense layer with ReLU activation, preceded by batch normalization.
4. The final layer is a dense layer with 1 neuron and sigmoid activation. This reflects the probability that a question pair is duplicate.

I have tried a few things to find a model with the best accuracy:

1. Number of training units (2/4/6/8).
2. Number of neurons in the dense layer (100/200/300).

3. Batch normalization (true/false).

This gives us 24 combinations to train the model on.

3.2.2 Loss and Optimization

Since this is a binary classification (duplicate or not), I have used binary cross entropy as a loss metric. To avoid the effort to train the hyper parameters by hand, I have used the Adam's optimizer.

3.2.3 Code Implementation

To build this neural network I have used Keras[12], which is a library that is built on top of Thanos[13] and Tensorflow[13]. I have used a single function that creates a neural network for every combination specified, thus I have built 24 neural networks and have trained them for 25 epochs (steps). The naming convention I have followed is *'no_neurons__batch_norm__no_units'*. Here is an example:

1. Number of training units – 2
2. Number of neurons in the dense layer – 200
3. Batch normalization – true/false

In the above case the model will be named '200_True_2'. The model will be saved to the memory as '200_True_2.h5' and the log of the training would be saved '100_True_2.h5.log'.

3.2.4 Order of questions in a pair

For my code implementation the order or the question pair matters. Hence to build an accurate model I pass the questions in the given and the reverse order. I then take the minimum of the two scores. I take the minimum so that the system gives a false negative instead of a false positive. For the application, false negative is acceptable but false positive is not.

3.2.5 Final Performance

The model that performed the best with an accuracy of 82.73% in my testing had the following parameters:

1. Number of training units – 6
2. Number of neurons in the dense layer – 300
3. Batch normalization – true

3.2.6 Performance of intermediate models

The below table shows the performance of the other 23 models.

Rows – number of layers

Columns – number of neurons in each layer

Batch normalization: off

	100	**200**	**300**
2	0.7838	0.7866	0.7847
4	0.7834	0.7808	0.7876
6	0.7853	0.7782	0.7818
8	0.7860	0.7758	0.7738

Batch normalization: on

	100	**200**	**300**
2	0.8130	0.8220	0.8231
4	0.8167	0.8221	0.8261
6	0.8174	0.8216	0.8278
8	0.8205	0.8244	0.8245

The following observations can be made:

1. The best performance of a model with batch normalization off is 78.76% while that with batch normalization on is 82.78%. Hence there is 4.02% improvement.
2. With batch normalization the lowest performance of a model on testing data is 81.30% an the highest is 82.78%. We can see that tuning of hyper parameters has improved the accuracy by 1.48%.

4 Results

4.1 Model Evaluation, Validation and Justification

The baseline model solution had an accuracy of 65.153% on the testing while the best model (82.73%) gave an improvement of 17.63% over the benchmark model that just compared bag of words and calculated the Jaccard index.

The final model had the following characteristics after 25 epochs:

- Training accuracy: 91.300%
- Validation accuracy: 82.370%
- Test accuracy: 82.783%

The training accuracy is higher than the validation/test accuracy, hence we can conclude that the model is slightly overfitting. Since the validation and test accuracy scores are close to each other, the model can classify unseen data well.

5 Conclusion

5.1 Free Form Visualization

The ability to recognize entities of question that are of interest in classifying the question is something the model should learn well to get good accuracy.
Consider the following questions from the dataset:

Question 1	What is the salary in India?
Question 2	What is the salary in Argentina?
Human opinion	Not duplicate
Model’s prediction	Not duplicate

The above 2 questions are not duplicates since they talk about salaries in different countries even though same keywords are used in them.

Consider the below questions which have the same meaning but are framed differently:

Question 1	What are some good places to visit in India
Question 2	Which are some tourist places in India
Human opinion	Duplicate
Model's prediction	Duplicate

Word vectors help the model recognize the basic intents behind each question and categorize similar words. By changing the name of the country in the question pair we see that they are no longer classified as duplicates.

Question 1	What are some good places to visit in India
Question 2	Which are some tourist places in Uzbekistan
Human opinion	Not duplicate
Model's prediction	Not duplicate

A drawback of using pre-trained word vectors, is that they can flag entities as similar is they are commonly used in the data used to train the GloVe.

Question 1	What are some good places to visit in Germany
Question 2	Which are some tourist places in France
Human opinion	Not duplicate
Model's prediction	Duplicate

Here, the model incorrectly classifies the questions as duplicate since in the training data of the GloVe Germany and France may have been used together and hence the GloVe understood both these are the same since they are European countries.

5.2 Reflection

The entire project was supervised learning, there was a target variable and some primitive features. But since this was NLP related, the features had to be extracted.

5.2.1 Feature extraction

This was one of the most important aspect of building a good model. NLP problems have text of variable lengths and machine learning algorithms need numeric values to learn. There were 2 methods that I discovered that could be used to convert this text numeric values, bag of words and word vectors. The reason I selected word vectors was because it could express similarity between words. The other necessity was getting a fixed length feature from the text. To fulfill this necessity I used max and mean pooling techniques. Once every question was converted to fixed length features, they were ready to be used by the model.

5.2.2 Training

Using deep neural networks in the NLP field is a popular convention. The topology of the network can have different combinations and to get the best accuracy I decided to train my model with all those combinations to find out what improvement those features can bring to the model. Batch normalization came to be a good techniques to improve the accuracy.

5.3 Improvement

5.3.1 LSTMs

There is a type of recurrent neural network called LSTM (long short term memory). Since the inputs to the network are of fixed size, I have used max and mean pooling to make this happen. But this results in loss of information. LSTMs allow to convert all the questions into comparable sequences of word vectors. A future work to this project could be to check if this would result in a higher accuracy model or not.

5.3.2 Training custom GloVe

Previously we saw that since the GloVe was pre-trained it considered Germany and France as one country, to avoid such problems we can train custom GloVe on the Quora dataset to get better results. However we will need way more data than what is currently available to be able to do that.

Bibliography:

[1] - https://stackoverflow.com/

[2] - https://answers.yahoo.com/

[3] - http://www.answers.com/

[4] – https://www.quora.com/

[5] - https://en.wikipedia.org/wiki/F1_score

[6] - https://en.wikipedia.org/wiki/Word2vec

[7] - https://nlp.stanford.edu/projects/glove/

[8] - https://spacy.io/usage/

[9] - https://stackoverflow.com/questions/37434426/max-pooling-vs-sum-pooling

[10] - https://en.wikipedia.org/wiki/Jaccard_index

[11] - http://jupyter.org

[12] - https://keras.io

[13] - http://deeplearning.net/software/theano/