Capstone Project 1

Duplicate detection in the Quora question pairs dataset

Zahiduzzaman Biswas

Abstract

Introduction

With the increasing access to the fast internet all over the world, internet has become one of the most popular sources of knowledge. The number of the users in the question-answer forums like Quora, Reddit, stackoverflow etc. is growing rapidly. Since millions of users, from different technical backgrounds and experiences, ask questions, similar or same questions, many a time, repeat multiple times. It eventually creates a maze of threads, where finding the best answer become a challenge for the new users. In order to provide the best and the most efficient user-experience, therefore, a reliable online knowledge base without any duplicate questions is essential. In the field of natural language processing, long standing researches are going on to build models that can automatically detect duplicate questions and combine them in one thread in order to help the users find the best possible answers to their questions. Recently, Quora hosted a competition on Kaggle.com, to find a better solution to the problem. Inspired by the competition, we made an attempt to build a machine learning mode that can efficiently detect the duplicate questions in the Quora question pair dataset.

In the competition, a training dataset with 404,290 labeled rows and a testing dataset of 3563475 question-pair rows without labels. Training dataset contains 2 text columns for the question pairs, 2 numeric columns of ids for each question, 1 numeric column for id of each question pair and 1 label column to indicate whether the pair is duplicate or not. In the training dataset, there is 1 column of test id for each question pair and 2 text columns for the question pairs.

The goal of this competition is to predict which of the provided pairs of questions contain two questions with the same meaning. A duplicate pair does not necessarily contain the same words. For example, “What is the step by step guide to invest in share market?” and “What is the step by step guide to invest in share market in India?” have almost same wording, but the intents of them are different; therefore, the are not duplicate. On the other hand, “How is the new Harry Potter book ‘Harry Potter and the Cursed Child’?” and “How bad is the new book by J. K Rowling?” do not have any word in common, but they are duplicate, for they have the same answer. Therefore, the challenge is whether the model can predict the questions in each pair are semantically equivalent and can be answered with the same words.

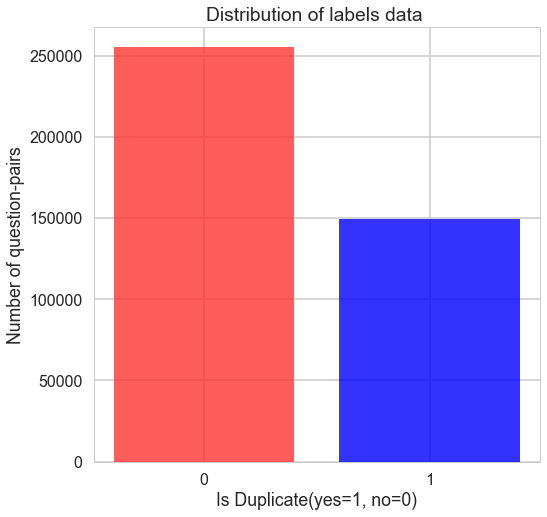
**Background/Related work**

1. Semantic similarity research is going on for long time. Wordnet is the result of the research of the Princeton.
2. Sentence similarity and word similarity are growing in importance
3. Recent work in the similar project by wu lang
4. Most of the projects in the Kaggle competition is notable. They mostly use Siamese artificial neural network

Approach:

1. Data exploration and preprocessing:

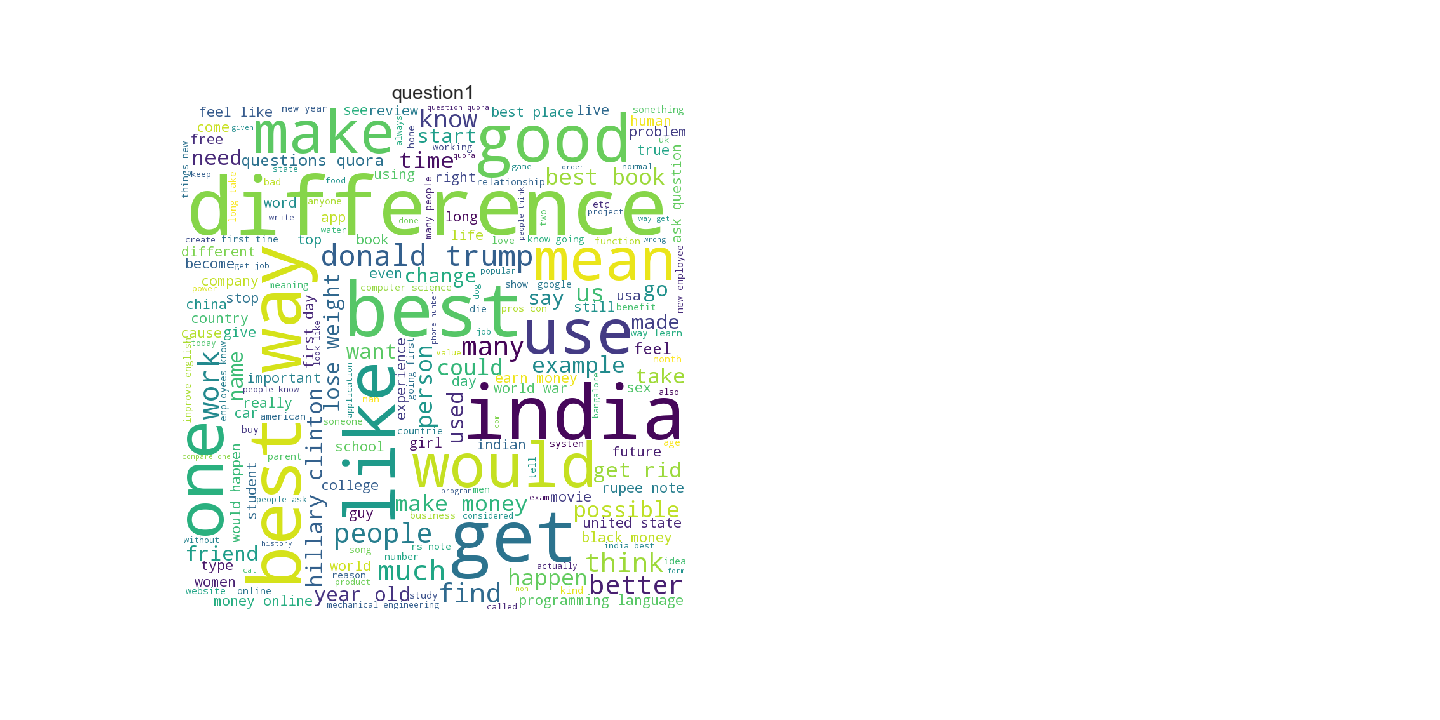
The provided training dataset is comparably clean, well-structured, and fully labeled. It has only two rows containing null values, which is less than 0.0005% of the overall training data. Therefore, these two were removed. Although many questions were repeated in each column but as a pair all the remaining 404,288 rows are unique. Finally, the training dataset has 149263 question pairs labeled as duplicate, and 255025 pairs are not duplicate.



**Figure 1: Bar plot of the labels in training dataset**

The test dataset, however, required some preprocessing. The provided dataset contains 3563475 rows of testing data, out of which about 262144 rows contains invalid test ids. Besides, the dataset also contains over a million of duplicate data. After cleaning the dataset, 2345796 unique rows are left for testing.

From the exploration of the remaining data, one of the most common words is found to be ‘india’, which indicates a large number of the questions were related to India. Some of the other most frequent words include ‘donald trump’, ‘hilary clinton’ etc. A good share of the questions, therefore, are asked about the USA presidential elections. Besides those, some words related to some trends such as ‘weight loss’, ‘programming language’, ‘make money’ etc. are clearly visible in the figure 2. Questions from different fields and perspectives are available in the training data, which adds complexity to the task of finding the common intents.

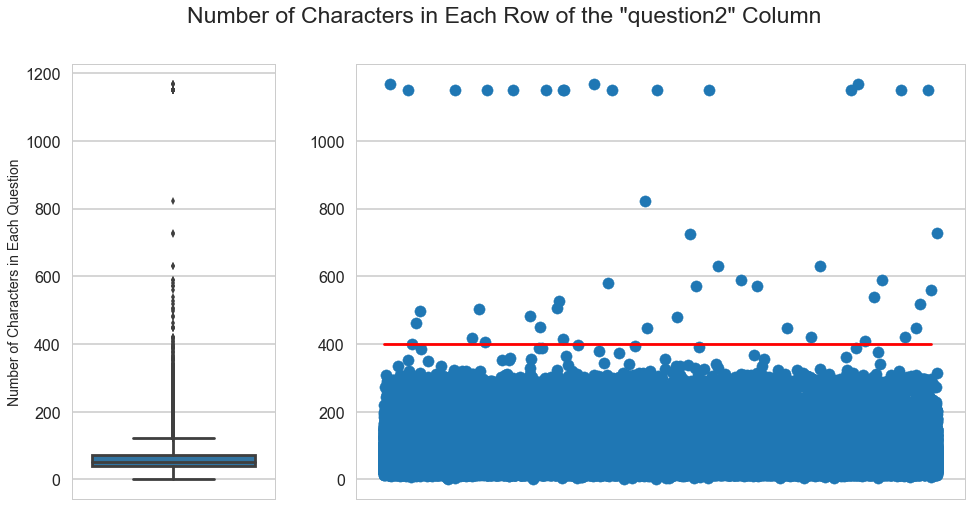
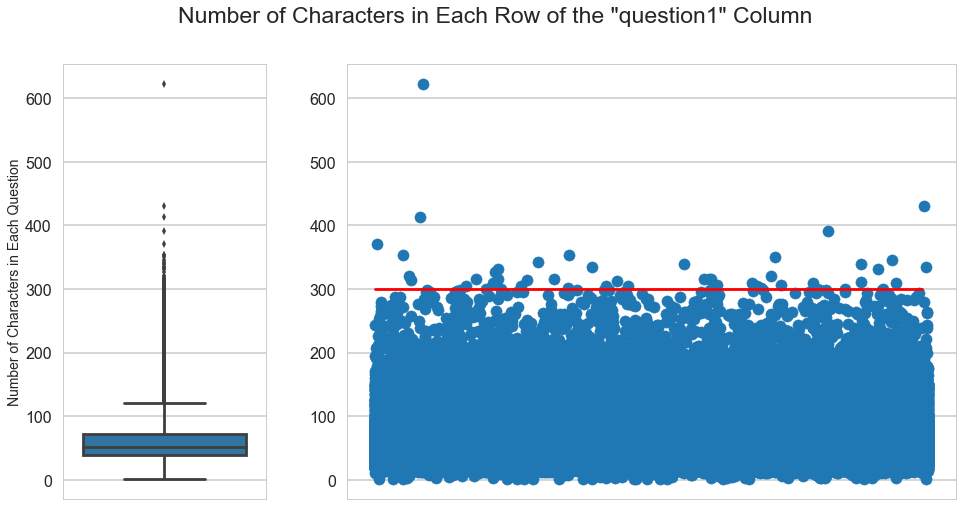


**Figure 2. Word cloud for most frequent words in the training dataset**

1. **Text preprocessing**

The purpose of the text preprocessing is to reduce each question text to a minimal form for efficient embedding and vectorization. A noisy text with redundant characters or words could hamper the accuracy of the model. Therefore, the following steps are performed at this stage:

* In the training dataset, most of the questions in question1 and question2 columns are consisted of less than 300 characters and less than 400 characters respectively (Firgure3). Any question exceeding those thresholds are counted as an outlier, and the corresponding rows are removed from the dataset.



**Figure 3: Plots of number of characters in each question**

* The question texts are at first converted to the lower case. All the non-alphanumeric characters such as punctuations, special characters, and foreign characters are then removed. Only the English letters, numbers, and spaces are left in each text.

Remove the Commonly used words (Stopwords)

Remove Non-Alphanumeric Characters

Convert to Lower Case

Vectorize text with Tf-idf vectorizer

Remove Multiple Whitespaces

Stem Texts to the Base Form

**Figure 4: Steps to the text data preprocessing**

1. Word embedding (bow, tfidf)

Bag of word representation

* 1. Tfidf representation

1. Sentence/Document Embeddings:

Doc2vec for sentence or document embedding

1. Semantic distances (cosine and Euclidean)
   1. Cosine distance between each question pair vectors
   2. Euclidean distance between each question pair vectors

Experiments:

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Models | Public Score (log\_loss) | Private Score (log\_loss) |
| 1 | Multinomial Naïve Bayes with default parameter,  Tf-idf tokenizer with default parameters | 0.438 | 0.440 |
| 2 | Multinomial Naïve Bayes with tuned parameter,  Tf-idf tokenizer with tuned parameters | 0.429 | 0.431 |
| 3 | Multinomial Naïve Bayes with default parameter,  Tf-idf tokenizer with default parameters,  Doc2Vec with default parameters trained on the Training question pairs | 0.398 | 0.399 |
| 4 | Multinomial Naïve Bayes with default parameter,  Tf-idf tokenizer with default parameters,  Doc2Vec with tuned parameters trained on the Training question pairs | 0.390 | 0.398 |
| 5 | Multinomial Naïve Bayes with default parameter,  Tf-idf tokenizer with default parameters,  Pretrained Doc2vec model on wikipedia | 0.404 | 0.405 |