Quora Question Pairs

Rakesh Singh Rawat
IIIT-Delhi
rakesh17056@iiitd.ac.in

Sharat Agarwal
IIIT-Delhi
sharata@iiitd.ac.in

1. Introduction

Question and answer websites likes Quora and Stack overflow allow their users to ask questions and other users to answer them. However the biggest challenge is the same question is being asked by many different users with different wording and sentence formation. This creates lots of duplicate questions for the same context and frustrates the users to search for the best answer of that question. Ideally these duplicate questions should be merged into a single canonical question by some ranking system, so that users can get the best answers to that particular question.

We say a pair of question is duplicate when the question askers express the same intent, i.e., valid answer to one question is also the valid answer for the other question. For example: “How can I be a good geologist” and “What should I do to be a great Geologist”. These two questions are duplicate and former one is the best possible representation among the two.

2. Related Work

Many research have been done in this area for a better extraction of words and sentences Bowman[1] used Neural Network based model that encoded words in the sentence. They encoded and concatenated two sentences and fed them in multilayer neural network for classification, this was the general framework which many people followed with some variation.

Rocktschel[2] introduced attention modeling that generate alignment between words of both sentences or entire sentences.

Parikh[3] modified Rocktschel by performing intra-sentence attention with 86.8% accuracy on textual dataset.

State of the art implementation till date is 86% accuracy on this Quora dataset, it has been reported that human accuracy on this dataset 86%.

3. Dataset and Evaluation

3.1. Dataset

We used the subset of dataset released by Quora that consist of over 400,000 lines of potential duplicate pairs. Due to computational limitations we have used 100,000 lines for training, validation and testing. Figure 1 shows the format of the raw dataset.

3.2. Dataset Evaluation

Figure 2 shows the ratio of duplicate to non-duplicate question pairs in original dataset compared to our train, test and validation set. Distribution is almost same so that we can map the results to original dataset. In 100,000 lines of training set there were 165931 total questions out of which 88% percent were unique and rest were repetitions. Also Figure 3 shows the histogram of word length for each question. There is one missing value in validation and test dataset. In addition while evaluating the dataset we found that writers express the questions in a very ambiguous manner with different symbols or either highly technical subjects. Lastly there are many questions with grammatical mistakes or frequently misspelled. It is worth noting that some questions have inherit ambiguity like, “How do I make 1000 rupees.” and “How can I make 1000 bugs” can be said duplicate if we assume that bugs means rupees.

Evaluation metric used are: precision, recall and F1 score. For evaluating state of the art model we are using cross-entropy.

4. Analysis and Progress

4.1. Word Share

Some question pairs share large proportion of words among them, logically they should be duplicate but they were found entirely different, so we did word share analysis among question pairs and we were surprised to conclude that large number of question pairs possessed this property. We thought that stop words were involved in contributing word matching but after their removal there was minimal difference in the observation Figure 4 illustrates the difference. For example:

1. "How to use pen effectively."
2. "How to use chainsaw effectively."

As this example clearly shows there are four words matching in a five word sentence but these two sentences are entirely different.

4.2. Baseline Models

Before jumping on to the state of the art techniques, we were curious about how the baseline model would perform on this data. The representation used for these baseline models is bag of words (BOW) of questions. For each question pair we take word vectors and concatenate two vectors to form a single sample for classifiers.

Finally we used two baseline approaches – Naïve Bayes and Random Forest classifier.

4.3. Word Embedding's

In this approach semantically similar words are mapped to nearby point. Figure 5 represents word embedding's trained from Kaggle dataset in form of plot. We have used Kaggle dataset word embedding vocabulary of 30 random words. This plot shows that semantically similar words are close to each other according to Kaggle dataset. In our final model, we will use pre-trained GloVe word vectors developed by Stanford from genism package.

These word embedding's will be input representation for sequence based LSTM architectures.

4.4 Challenges

After exploring the dataset the biggest challenge is to solve the ambiguity in the sentences as stated above in section 3.2. Also the baseline models which we have used are not able to capture to semantic property of the sentence so to overcome this we have to use sequence based model, adding to it the dataset it bit computationally complex and we lack in resources for high computational power. Also the ranking criteria for duplicate question will be bit challenging as selecting the appropriate question according to the best and appropriate word used in the sentence is what need to be thought of.

5. Results

The accuracy obtained from baseline models naïve bayes and random forest are 72 and 77% percent respectively which is pretty good, but these techniques are not utilizing semantics of the questions. State of the art techniques gives far better results than these models. In bag of words approach, we are simply giving an id to the words, to convert them into number representation. Results obtained from baseline models are provided in Table 1. Figure 6 shows the confusion matrix for testing data on naïve bayes and random forest.

One improvement over this approach can be to utilize semantics by using Word Vectors, which capture the semantics of each word in the sentence in from of vectors.

6. Future Work

From the above analysis, it is clear that we will be highly benefited by using sematic similarity of questions for training our model. For remaining semester, we will try Long Short Term Memory (LSTM) RNN, with word by word attention, distance and angle.

Word Embedding's will be fed into LSTM cells and then combined using different architectures. Figure 7 illustrates the sample LSTM architecture.

Tasks Remaining:

(1) Implementing LSTM architectures in keras, as these are the state of the art techniques. (Rakesh)

(2) Analysis from combination of different hyper parameters on LSTM model and deciding best parameters. (Sharat)

(3) Implementation and collection of results from Ranking criteria (Sharat)

(4) Analysis of Ranking criteria on data labelled by model.(Rakesh)

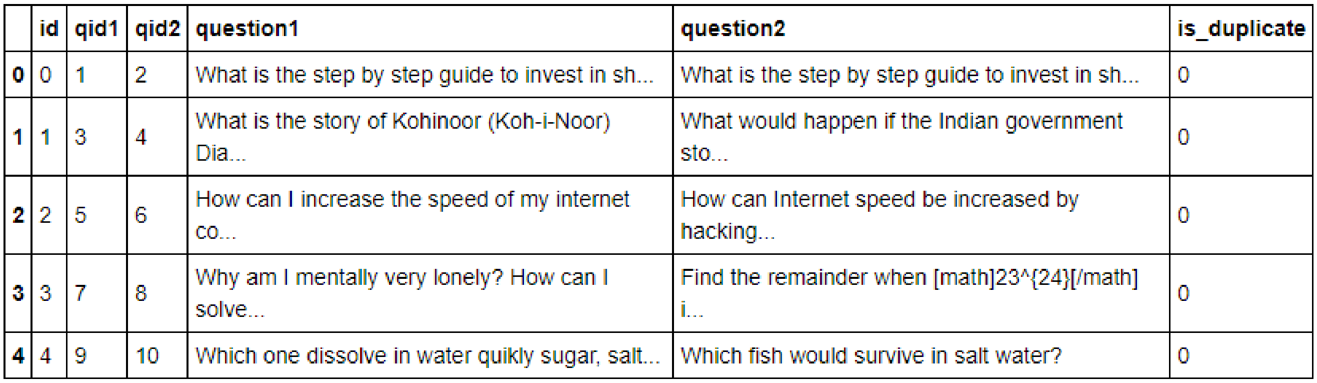
References

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts and Christopher D Manning. A large annotated corpus for learning natural language inference, In Conference on Empirical Mehods in Natural Language Processing.

[2] Tim Rocktaschel, Edward Grefenstete, Karl Hernann, Tomas Kocisky and Phil Blunson. Reasoning about Entailment with Neural Attention. In Conference on Empirical Methods in Natual Language Processing (EMNLP), 2015.

[3] Ankur P. Parikh, Oscar Tackstorm, Dipanajn Das, and Jakob Uszkoreit. A Decomposable Attention Model for Natural Language Inference. In Conference on Empirical Methods in Natural Language Processing (EMNLP), 2016.

Figures and Tables



	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

Figure 1: Raw Data Format

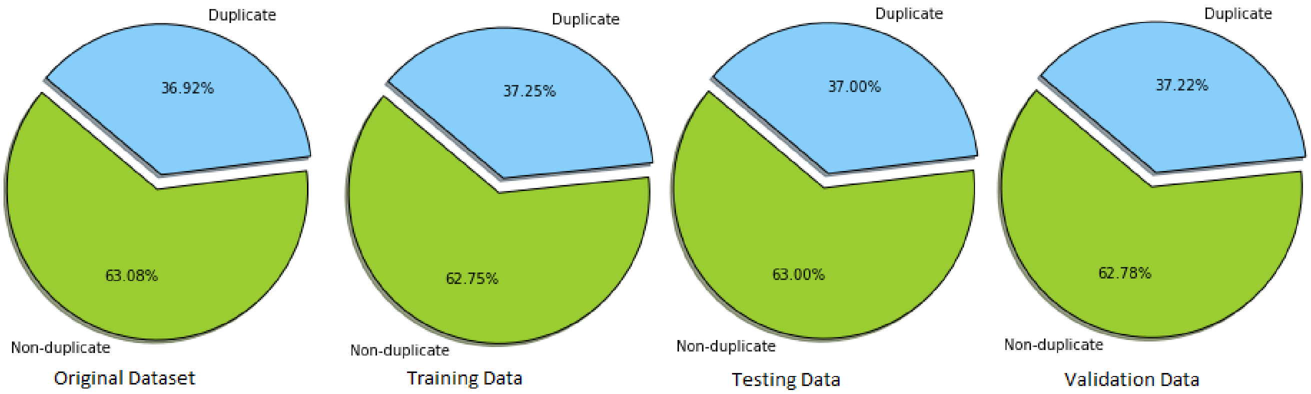


Figure 2: Ratio of duplicate to non duplicate questions

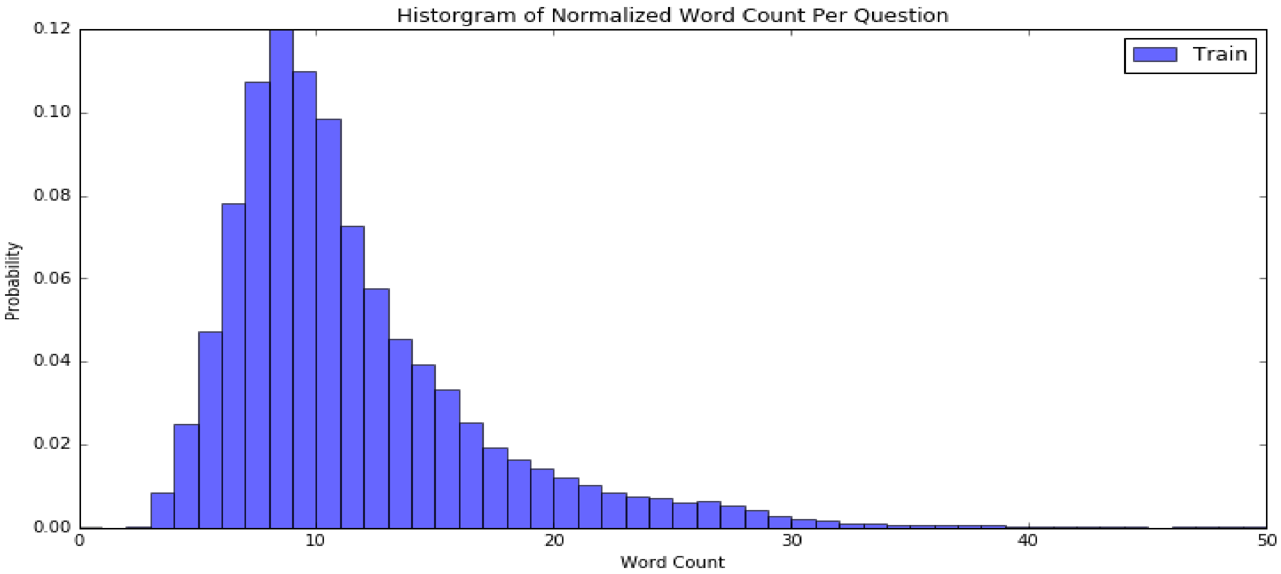


Figure 3: Histogram of Normalized Word Count per Question

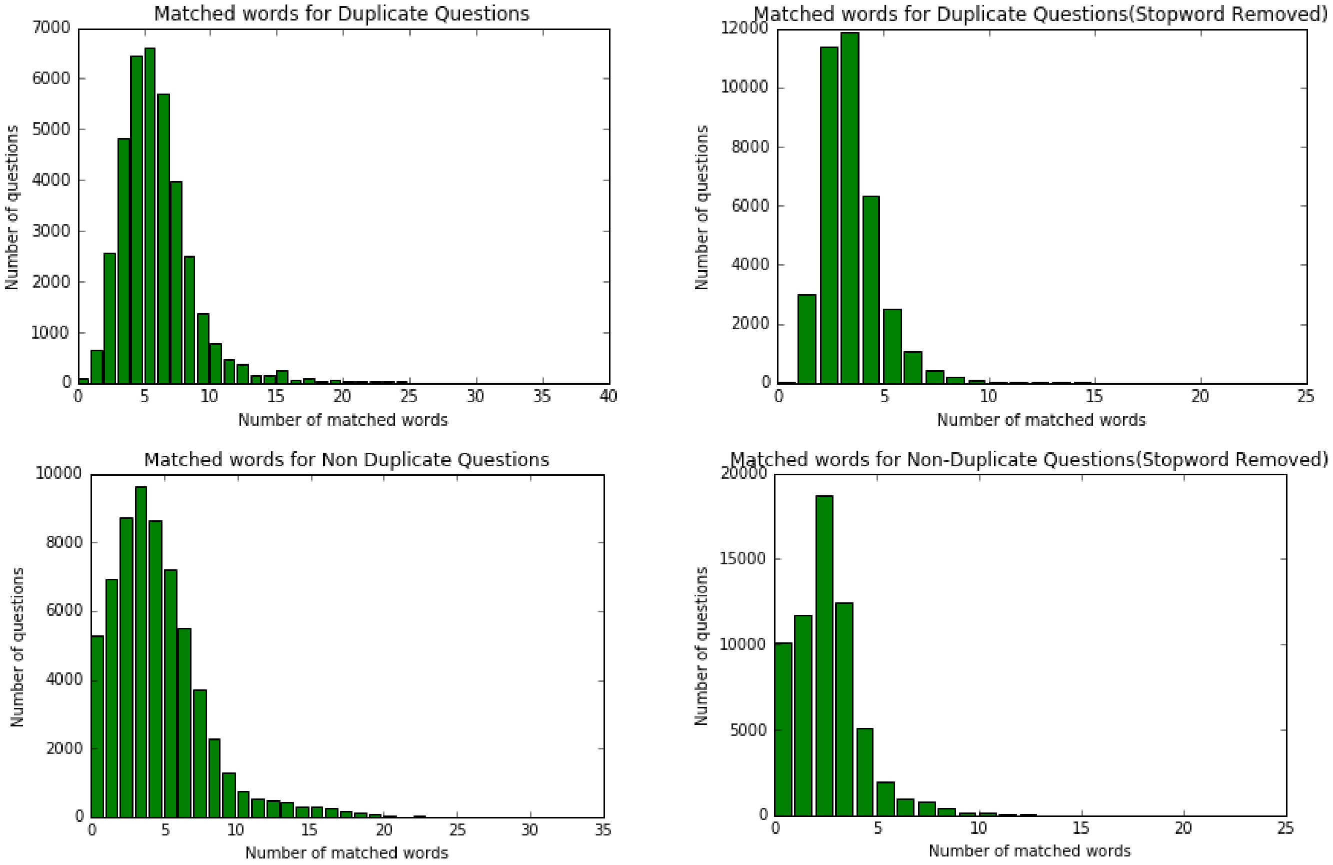


Figure 4: Graphs illustrating the difference in matched words before stopword removal and after stopword removal

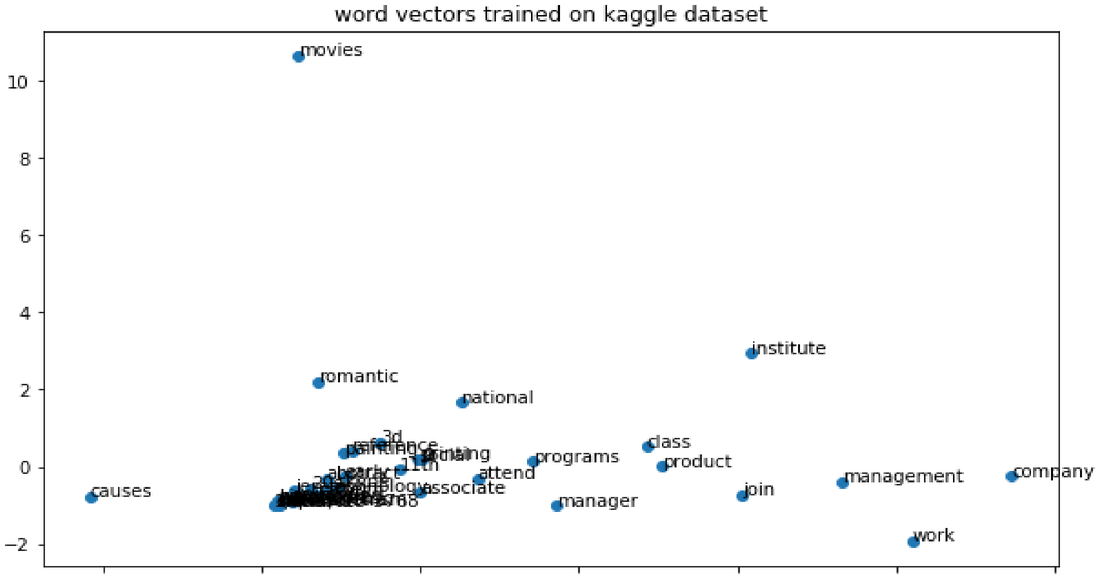


Figure 5: Word Vectors trained On Kaggle Dataset

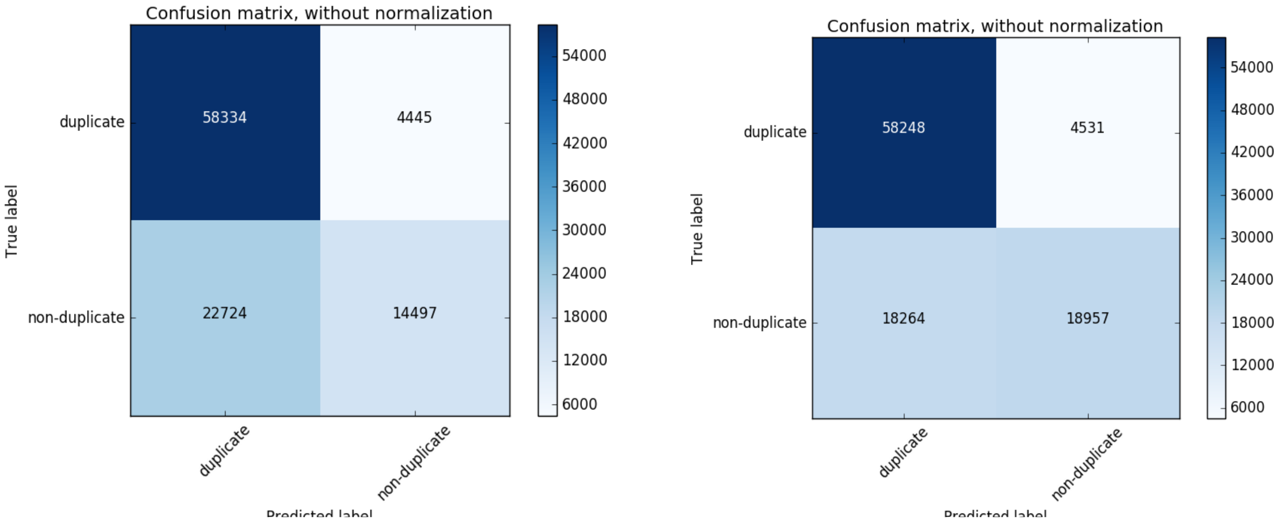


Figure 6: Confusion matrix naïve bayes and random forest on testing data.

	Naïve Bayes	Random Forest
Accuracy	0.728310	0.772050
Precision	0.765336	0.807093
Recall	0.389484	0.509309
F1 Score	0.516247	0.624520

Table 1: Evaluation metric for test data

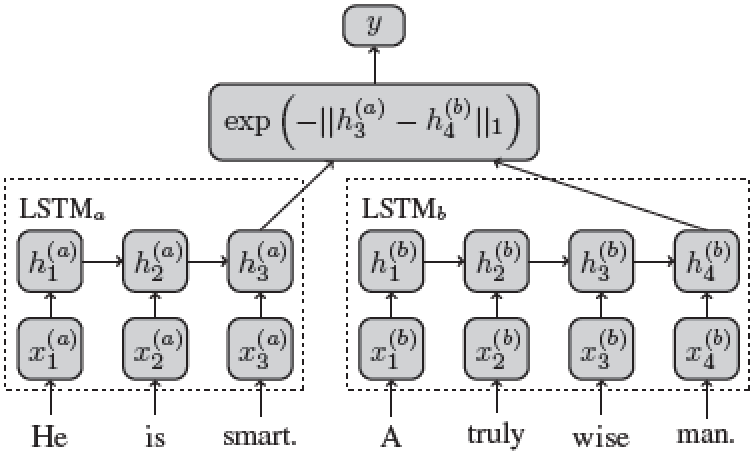


Figure 7: Sample of LSTM architecture