Identification of Duplication in Questions Posed on Knowledge Sharing Platform Quora using Machine Learning Techniques

R. Rishickesh, R.P. Ram Kumar, A.Shahina, A. Nayeemullah Khan

Abstract— Quora, an online question-answering platform has a lot of duplicate questions i.e. questions that convey the same meaning. Since it is open to all users, anyone can pose a question any number of times this increases the count of duplicate questions. This paper uses a dataset comprising of question pairs (taken from the Quora website) in different columns with an indication of whether the pair of questions are duplicates or not. Traditional comparison methods like Sequence matcher perform a letter by letter comparison without understanding the contextual information, hence they give lower accuracy. Machine learning methods predict the similarity using features extracted from the context. Both the traditional methods as well as the machine learning methods were compared in this study. The features for the machine learning methods are extracted using the Bag of Words models- Count-Vectorizer and TFIDF-Vectorizer. Among the traditional comparison methods, Sequence matcher gave the highest accuracy of 65.29%. Among the machine learning methods XGBoost gave the highest accuracy, 80.89% when Count-Vectorizer is used and 80.12% when TFIDF-Vectorizer is used.

Keywords—Quora Question Pairs, Count-Vectorizer, TFIDF-Vectorizer, Machine Learning Methods

I. INTRODUCTION

Quora is an online platform to ask and answer questions across a community of users, who express their opinions to the questions posted. Founded in 2009, over the years it has grown leaps and bounds among the online community. Since people are entitled to post their questions and opinions, Quora is flooded with millions of questions with many of them being unanswered too. Most of the questions that are left unanswered have duplicates which are not taken care of. The new questions ought to be compared with the existing ones to check the similarity between both and if they convey the same meaning then the new one must be discarded. Traditional comparison methods like Sequence matcher and Cosine similarity involve simple implementations [1], [2], [3]. They check the sentences letter by letter. They count the number of characters that do match each other in the same position in both the sentences and will output a coefficient which is checked with a threshold value (1). If it is equal to 1 then the output is also equal to 1(sentences are duplicates of each other) and if it is less than 1 then the output is 0(sentences are not duplicates of each other).The drawback of using the methods discussed above are they give a very low accuracy for example, sentences “How old are you?” and “What is your age?” convey the same meaning and are duplicates of each other. But both Cosine similarity and Sequence matcher will classify them as sentences conveying different meaning.

To overcome this problem, Machine Learning (ML) methods are used [4], [5], [6] after applying Natural Language Processing (NLP) techniques like removal of stop words, stemming and lemmatizing the sentences. The sentences are then converted into vectors and fed into the respective models to predict the output.

Zihan Chen et.al, proposed a Siamese RNN-LSTM network which was further developed using Manhattan LSTM model as a reference [7]. The new Siamese-LSTM model, consisted of 2 identical LSTM sub networks along with a 3-layer neural network to classify the duplicate set of questions. Yushi Homma et.al proposed a Siamese-GRU network [8] approach to encode the sentences. A wide range of distance measures were used to predict the similarity between the questions. Results suggest that while logistic regression with pure distance measures gave a decent performance, concatenating the transformed outputs and sending them to a neural network significantly improves the performance.

This article discusses both traditional comparison methods and machine learning methods to show the efficiency of the machine learning models over the traditional comparison methods. These models are considered over the Siamese networks because they are computationally more efficient by performing faster than the Siamese networks at the same time they perform equally with the respect to the Siamese networks..

Revised Manuscript Received on October 30, 2019.
* Correspondence Author

R. Rishickesh*, Department of Information Technology, SSN College of Engineering, Kalavakkam-603110, India

R.P. Ram Kumar, Department of Information Technology, SSN College of Engineering, Kalavakkam-603110, India

A. Shahina, Department of Information Technology, SSN College of Engineering, Kalavakkam-603110, India .

A. Nayeemulla Khan, School of Computing Science and Engineering, VIT University, Chennai-600127, India

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

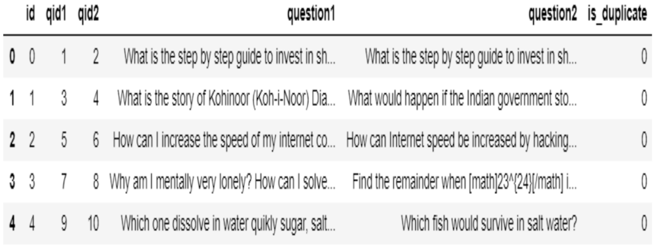
The accuracies of the models are evaluated and compared with each other and the best performing model is found. The models considered for traditional comparison are Cosine similarity [1], [2] and Sequence matcher [3]. Supervised learning models like Naïve Bayes Classification, Support Vector Machine (SVM) and logistic regression are considered for modeling in addition to the deep learning models like Shallow neural network and Recurrent Neural Networks (RNN). Bag of Words (BOW) model [9] is used for the vector representation of the textual context, which is fed as the input to the machine learning models. The 2 types of BOW models used in the present work are Count-Vectorizer and TFIDF-Vectorizer.

II. DATASET

The dataset used in the present study is the "Quora Question Pairs" is given by Kaggle. The dataset was officially released by the Quora Machine Learning team in their website in 2017.

III. EXPLORATORY DATA ANALYSIS

The dataset consists of the following features: the unique identification number for each question pair, the unique identification for each question present in the table, both the questions in their full text format and the target variable which indicates whether both them are similar or not (if it is 1 then the questions are similar less they are dissimilar).



	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

Fig.1 Represents first 5 rows of the training dataset.

The data folder present in the kaggle "Quora question pairs" problem contains two csv files. One is the 'train' dataset and the other is a 'test' dataset. The 'train' dataset is the one on which we will be modeling as it contains the target variable (is_duplicate). The train dataset contains 6 columns 'id', 'qid1', 'qid2', 'question1', 'question2' and 'is_duplicate' as shown in the Fig.1. The 'id' column uniquely identifies each question pair in the dataset. The 'question1' and 'question2' columns are sentences on which the similarity has to be checked. The 'is_duplicate' column indicates whether the two sentences are duplicates of each other. There are totally 404290 question pairs in the 'train' dataset and 537933 questions in the 'train' dataset.

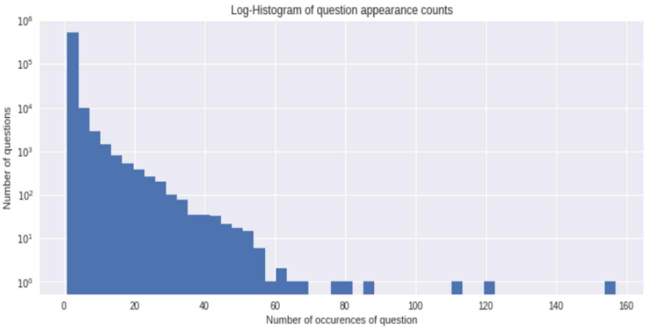


Fig.2 Histogram represents the question appearance counts in the 'train' dataset.

There are many questions that appear regularly in the dataset. The number of questions that appear only once in the dataset is over 500000 as shown in the Fig.2. There is a question that appears around 158 times. There are about 48 characters that constitute almost 2% in the question pairs across the 'train' dataset and almost 50 characters constitute about 1.8% in the 'test' dataset as shown in Fig.3. The minimum length of a question in both the 'question1' and 'question2' columns is 1. The number of appearances of such questions in 'question1' column is 67 and the number of appearances in the 'question2' column is 24.

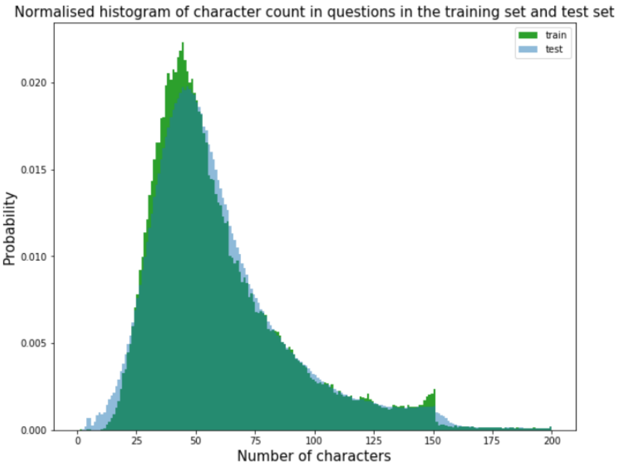


Fig.3 Histogram represents the character count in both 'train' and 'test' dataset.

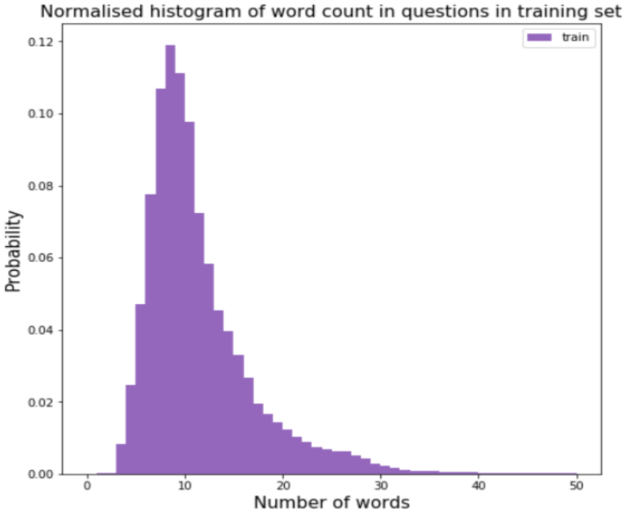


Fig.4 Histogram represents the word count in 'train' dataset.

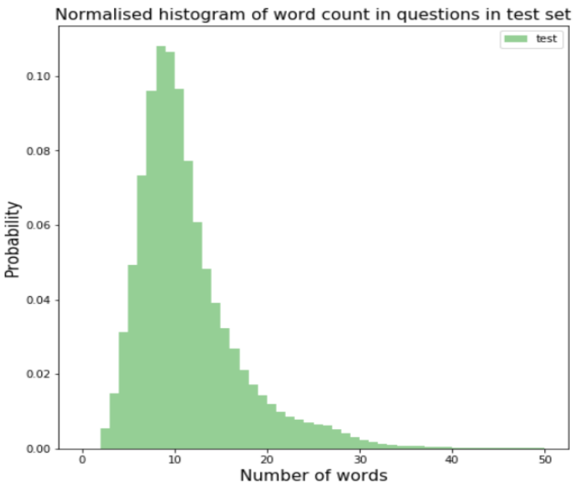


Fig.5 Histogram represents the word count in 'test' set.

Around 8 words occur the more frequently (almost 12% of available the words in the dataset) than the rest of the words in the 'train' dataset as shown in Fig.4.

Those set of words include *is, a, the* which belongs to the set of stop words that has no significant meaning regarding a question. Similarly, around 9-10 words have the most occurrences in the 'test' set as shown in Fig.5. The average number of common words is higher between similar sentences as shown in Fig.6.

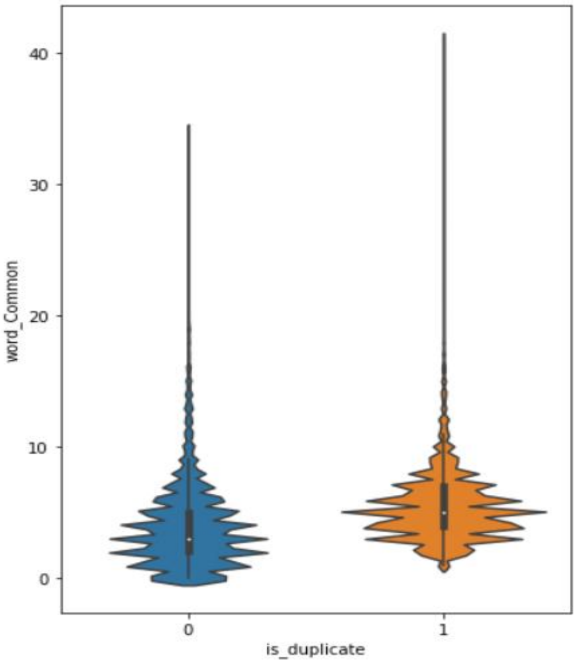


Fig.6 Violin Plot depicting variation in the number of common words for both similar and non-similar questions.

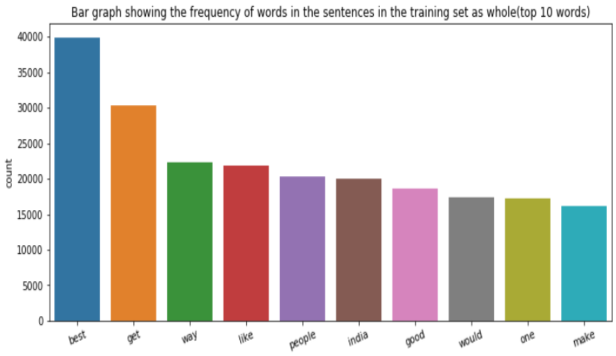


Fig.7 Bar graph showing the frequency of words in the sentences in the 'train' set(top 10 words)

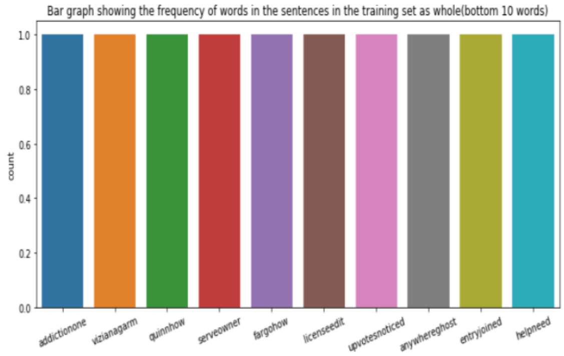


Fig.8 Bar graph depicting the frequency of words in the sentences in the 'train' dataset (bottom 10 words)

The most frequently appearing non-stop word in the 'train' dataset is the word *best* as shown in the Fig.7. It appears about 40000 times in the dataset. The next in the list of most frequently occurring non-stop word is *get* that appears well over 30000 times. The least frequently occurring words in the dataset are *serveowner*, *upvotesnoticed*, etc as shown in Fig.8. They occur only one time in the dataset.

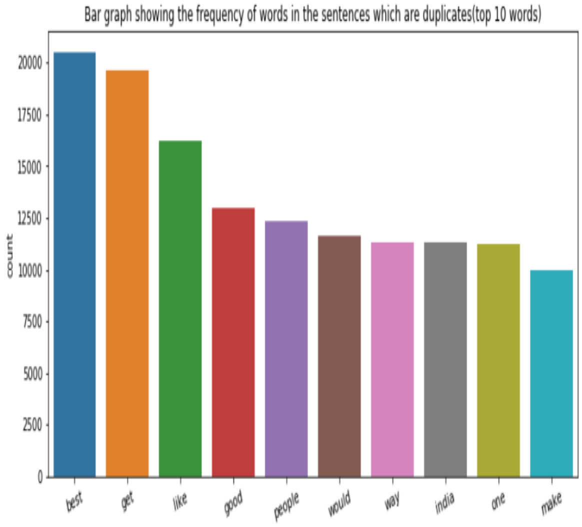


Fig.9 Bar graph showing the frequency of words in the sentences which are similar(top 10 words)

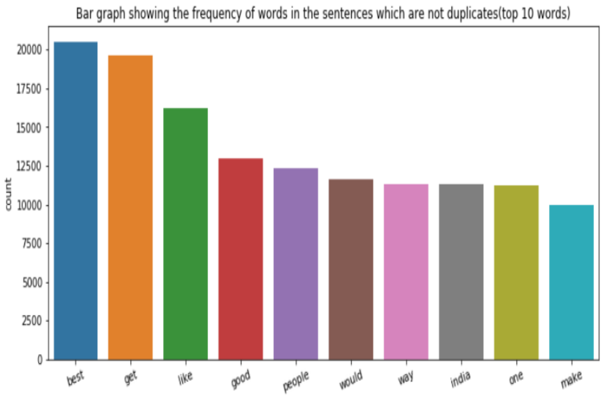


Fig.10 Bar graph depicting the frequency of words in the sentences which are not similar(top 10 words)

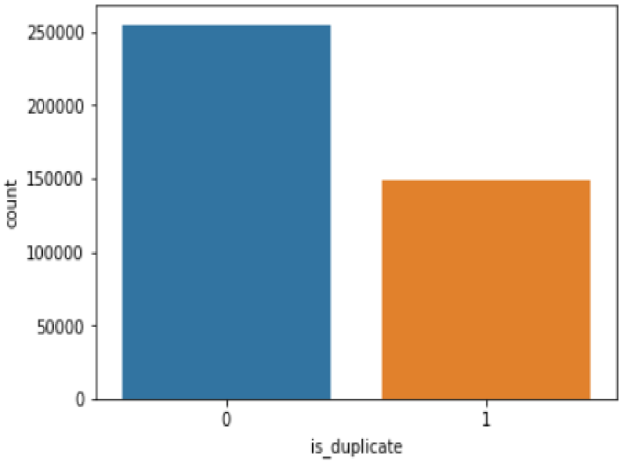


Fig.11 Bar graph illustrating the frequency of sentences that are similar and dissimilar.

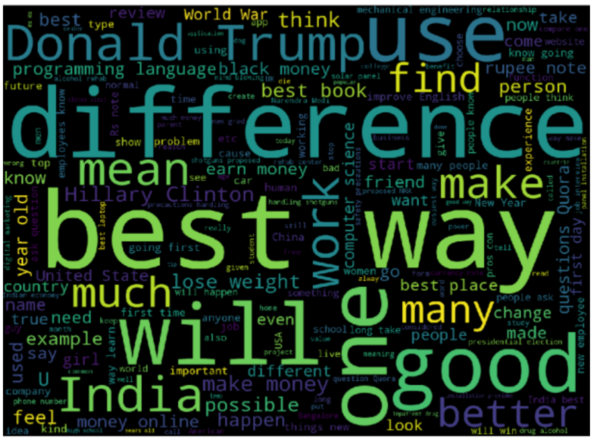
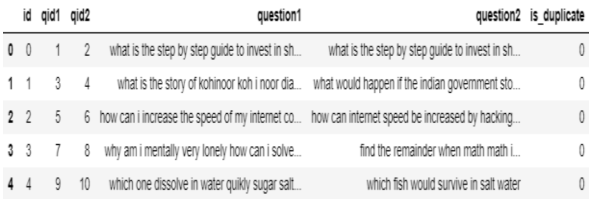


Fig.12 Word cloud of the ‘train’ dataset which shows the commonly occurring words in the sentences.

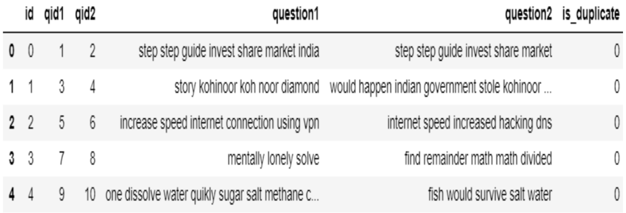
The number of question pairs that are similar to each other in the dataset are 149263 and the number of question pairs that are not similar to each other is 255024 as shown in the Fig.11. The ratio of number of questions based on their similarity is almost 1.7:1 . The most commonly occurring word in the question pairs that are similar is the word *best* followed by *get* as shown in the Fig.9. For the question pairs which are not similar, the commonly occurring word is also *best* as shown in the Fig.10. Proper nouns like *Donald Trump*, *India* also feature frequently in the questions as shown in the Fig.12.

IV. DATA PRE-PROCESSING



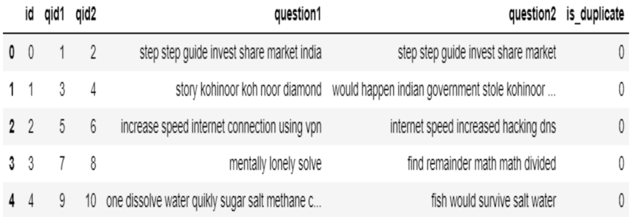
	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	what is the step by step guide to invest in sh...	what is the step by step guide to invest in sh...	0
1	1	3	4	what is the story of kohinoor koh i noor dia...	what would happen if the indian government sto...	0
2	2	5	6	how can i increase the speed 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 math i...	0
4	4	9	10	which one dissolve in water quikly sugar salt...	which fish would survive in salt water	0

Fig.13 Represents the ‘train’ dataset having removed the punctuation marks and also converting all upper case characters to lower case.



	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	step step guide invest share market india	step step guide invest share market	0
1	1	3	4	story kohinoor koh noor diamond	would happen indian government stole kohinoor ...	0
2	2	5	6	increase speed internet connection using vpn	internet speed increased hacking dns	0
3	3	7	8	mentally lonely solve	find remainder math math divided	0
4	4	9	10	one dissolve water quikly sugar salt methane c...	fish would survive salt water	0

Fig.14 ‘Train’ Dataset after removing the stop words from both ‘question1’ and ‘question2’ columns.



	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	step step guide invest share market india	step step guide invest share market	0
1	1	3	4	story kohinoor koh noor diamond	would happen indian government stole kohinoor ...	0
2	2	5	6	increase speed internet connection using vpn	internet speed increased hacking dns	0
3	3	7	8	mentally lonely solve	find remainder math math divided	0
4	4	9	10	one dissolve water quikly sugar salt methane c...	fish would survive salt water	0

Fig.15 ‘train’ Dataset after performing lemmatization on both ‘question1’ and ‘question2’ columns.

The dataset is preprocessed by applying the following techniques:-

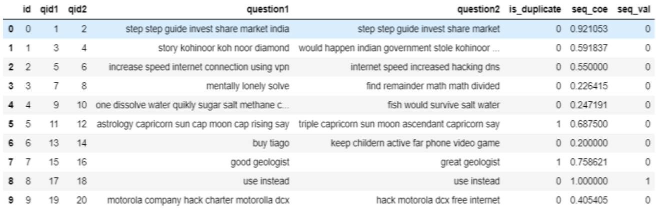
1) Removal punctuation marks and converting upper case to lower case characters.
2) Removal of stop words from the two columns
3) Lemmatization of the two columns.

The dataset is first preprocessed by removing the punctuation marks and converting all the upper case characters to lower case characters to remove the case sensitiveness. Fig.13 shows the dataset after the removal of all the punctuation marks and case conversion. Then the dataset, as shown in Fig.14, is processed by removing the stop words like *the, is, a* as those words do not convey any significance to the questions. Then the words in the dataset are lemmatized where the words suffices are removed are converted to the common noun form as shown in the Fig.15. For example words like *Cars*, *Cars’* are converted to *Car*. The dataset ‘train’ is split in the ratio of 70:30 for the training and testing samples respectively. The samples are then used for machine learning models to predict the similarity between questions.

V. TRADITIONAL COMPARISON TECHNIQUES

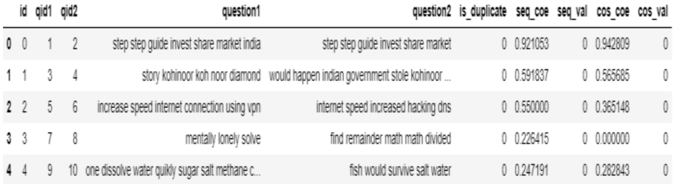
The traditional comparison techniques that were used to measure the accuracy in finding duplicate sentences are:-

i) Sequence matcher
ii) Cosine similarity



	id	qid1	qid2	question1	question2	is_duplicate	seq_coe	seq_val
0	0	1	2	step step guide invest share market india	step step guide invest share market	0	0.921053	0
1	1	3	4	story kohinoor koh noor diamond	would happen indian government stole kohinoor ...	0	0.591837	0
2	2	5	6	increase speed internet connection using vpn	internet speed increased hacking dns	0	0.550000	0
3	3	7	8	mentally lonely solve	find remainder math math divided	0	0.226415	0
4	4	9	10	one dissolve water quikly sugar salt methane c...	fish would survive salt water	0	0.247191	0
5	5	11	12	astrology capricorn sun cap moon cap rising say	triple capricorn sun moon ascendant capricorn say	1	0.687500	0
6	6	13	14	buy tiago	keep childern active far phone video game	0	0.200000	0
7	7	15	16	good geologist	great geologist	1	0.758621	0
8	8	17	18	use instead	use instead	0	1.000000	1
9	9	19	20	motorola company hack charter motorolla dcx	hack motorola dcx free internet	0	0.405405	0

Fig.16 dataset after creating two new columns ‘seq_coe’ and ‘seq_val’ to find similarity using Sequence matcher.



	id	qid1	qid2	question1	question2	is_duplicate	seq_coe	seq_val	cos_coe	cos_val
0	0	1	2	step step guide invest share market india	step step guide invest share market	0	0.921053	0	0.942809	0
1	1	3	4	story kohinoor koh noor diamond	would happen indian government stole kohinoor ...	0	0.591837	0	0.565685	0
2	2	5	6	increase speed internet connection using vpn	internet speed increased hacking dns	0	0.550000	0	0.365148	0
3	3	7	8	mentally lonely solve	find remainder math math divided	0	0.226415	0	0.000000	0
4	4	9	10	one dissolve water quikly sugar salt methane c...	fish would survive salt water	0	0.247191	0	0.282843	0

Fig.17 dataset after creating two new columns ‘cos_coe’ and ‘cos_val’ to find similarity using Cosine similarity.

A. Sequence matcher

Sequence matcher is used to find the Longest Contiguous Matching Subsequence (LCMS) between any two strings that do not have any junk elements [3]. Then the ratio of similarity between the two sentences (which lies within the range of [0,1]). The total sum of all matched sequences id found the ratio is calculated using the formula:-

$$ratio = 2 * \frac{M}{T} \qquad (1)$$

Where, *M* is the number of matches and *T* is the total number of elements in both sequences.

The threshold value for the ratio is 1.0 i.e. if the ratio value between the two strings is equal to 1.0 then similarity value between the 2 strings is assigned as 1 and if the ratio is less than 1.0 then the value is assigned the value of 0. In the dataset the column ‘seq_coe’ and ‘seq_val’ represent the similarity ratio and similarity coefficient, respectively as shown in Fig.16.

If the value is 1 in the ‘seq_val’ then the indication is that the questions are similar in meaning and if it is 0 then the sentences are not similar.

B. Cosine Similarity

Cosine similarity is the similarity between any 2 vectors that is found using the dot product between the 2 vectors [1], [2]. In case of sentences, the vector representation of 2 sentences is taken and the dot product is calculated. The vector representation of each sentence consists of the words' frequencies. The dot product of the 2 vectors is calculated using the formula,

$$c = \vec{a}.\vec{b} \qquad (2)$$

The similarity ratio between the 2 sentences is calculated using the formula,

$$ratio = \frac{\vec{a}.\vec{b}}{|\vec{a}| * |\vec{b}|} \qquad (3)$$

The threshold value for the ratio is also taken as 1.0 i.e. if the ratio value between the two strings is equal to 1.0 then similarity value between the 2 strings is assigned as 1 and if the ratio is less than 1.0 then the value is assigned the value of 0. In the dataset the column ‘cos_coe’ and ‘cos_val’ represent the Similarity ratio and similarity coefficient respectively as shown in Fig.17. If the value is 1 in the ‘cos_val’ column then the indication is that the questions are similar in meaning and if it is 0 then the sentences are not similar.

VI. MACHINE LEARNING TECHNIQUES

A. Naïve Bayes Classifier

Naïve bayes is a supervised classification technique based on the bayes algorithm [10], [11]. The model considers the features of the class to be independent and with zero correlation among themselves. Bayesian model is very useful for large datasets like the “Quora Question pairs dataset”. The algorithm is used to calculate the probability $P(A|B)$ using the formula:-

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \qquad (4)$$

For example, a racquet is termed to be a table-tennis racquet if it is small and is made by the combination of wood and rubber. Though the features may look to be dependent, the model does not take their dependencies into consideration.

B. Artificial Neural Networks (ANN)

An ANN is a computational model for information processing that involves a network of simple processing elements, called neurons [12], [13]. They are influenced by the structure of human brain where the artificial neurons present in the ANN perform functions similar to the biological neurons present in the brains. The neurons transmit signals and information using the artificial synapse similar to how a biological neuron transmits signals to various parts of the brain using the synapse.

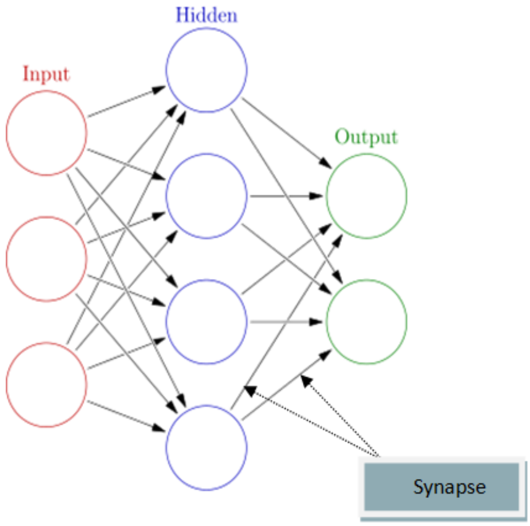


Fig.18 Structure of a feed foward ANN

The network consists of hidden layer in addition to the input and output layer as shown in the Fig.18. The input layer does not perform any processing, its main objective is to forward the input to the hidden layer [14]. The hidden layers are the places where the real time processing of data is done. An ANN can have more than 1 hidden layer.

In the present a 3 layer shallow neural networks is used. The shallow neural networks have very few hidden layers and are generally feed forward in nature.A neuron's function is to take the weighted sum as its input and output the value 1, if the sum is greater than some threshold value otherwise output the value 0. The threshold comparison is done in a neuron exclusively by the activation function.

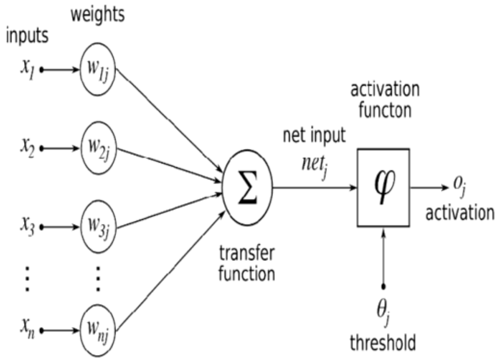


Fig.19 Structure of an artificial neuron.

The weighted inputs are sent to the transfer function where the inputs are mathematically summated and sent to the activation function as shown in Fig.19. Finally the activation function computes the output value by comparing it with the threshold value. The different types of activation function available are step functions, binary sigmoid function, bipolar sigmoid functions and softmax function. The formula for activation function is:-

$$f\left(b+\sum_{i=1}^{n} x_i w_i\right) \quad (5)$$

Where,

b is the bias

x is the input to neuron

w is the weight

n is the number of inputs from the incoming layer

C. Recurrent Neural Networks (RNN)

RNN are a variation of ANN where the connections between the neurons form a cycle similar to a graph where the output of the present state is fed as an input to the future state [15], [16]. The networks consist of memory elements that store the outputs and process them for future decisions. This enables RNNs to exhibit dynamic characteristics and helps them to make temporal decisions. The mathematical formulation to describe this memory function is:-

$$h_t = \phi(Wx_t + Uh_{t-1}) \quad (6)$$

Where,

t is the time step

h_t is the hidden state of the network at time *t*

W and *U* are the constant weight matrices

x_t is the input function

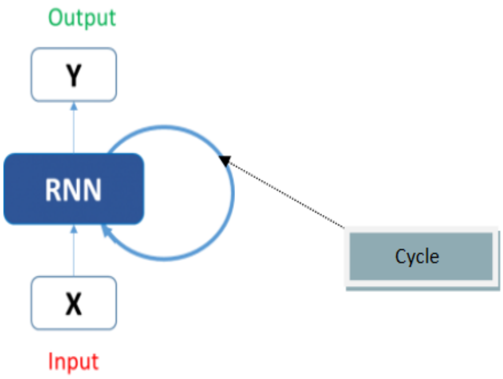


Fig.20 Represents the typical structure of a RNN

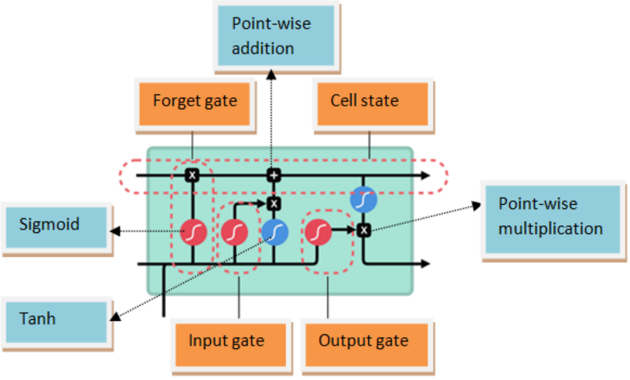


Fig.21 Represents the typical structure of a Long Short Term Memory

The output of the RNN is fed back as in input into the neural network for future usage as shown in Fig.20. The drawback of RNN is that it stores information only for a short span of time after which the information is lost. Long Short Term Memory (LSTM) is a memory element to overcome this problem. LSTMs [17] are composed of 3 gates: forget gate in addition to the input and output gate as shown in Fig.21. The LSTMs stores the values over a random time intervals and flow of information is supervised by the 3 gates. LSTMs also solve the vanishing gradient problem that is frequently found in RNN.

D. Logistic Regression

Logistic regression is a supervised learning algorithm used when the dependent variable is categorical, preferably binary [18], [19]. The logistic regression uses a logistic function, which is also called a sigmoid function. The function outputs a value from 0 to 1 when an input 'z' is fed into it. The

sigmoid function is as follows:

$$\phi(z) = \frac{1}{1+e^{-z}} \quad (7)$$

The sigmoid function outputs the probability of the occurrence of '*z*'

E. Support Vector Machine (SVM)

The main objective of SVM algorithm is to find the hyperplane which among the other possible hyperplanes classifies the data uniquely and correctly in the N-dimensional space [20], [21].

The hyperplane found has a maximum width i.e. maximum distance or width between data points of both classes in the dimensional space.

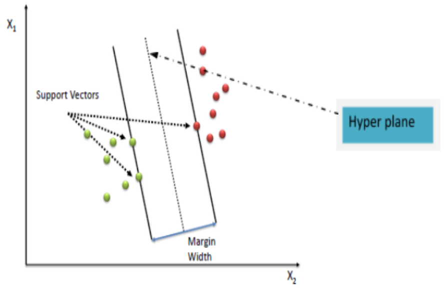


Fig.22 Graphical representation of hyper plane and the support vectors.

Fig.22 represents a hyperplane separating data points into two groups. The hyperplane's dimensions are proportional to the number of features present in the dataset 'N' ('N' can be any integer from 2). Higher the number of features higher will be the complexity of finding the right hyperplane.

SVMs use kernels to transform an input data to another form [22]. The functions are of the types – Linear, Non-linear, Polynomial, Sigmoid and Radial Basis Function (RBF). In the present work RBF is used as the SVM kernel.

F. Boosting classifier

Boosting algorithms are machine learning algorithm which is an ensemble of decision trees. They are primarily used to reduce the bias and to improve accuracy of a model [23]. Boosting algorithm's objective is to create a strong classifier from a set of weak classifiers in a sequential manner. Strong classifiers are built by building an initial model from the training dataset.

Then an improvised second model is built minimizing the errors found in the first model. Models are further added until the training set is correctly fit or once the saturation state of the model's accuracy is reached. The most commonly used boosting algorithm is the XGBoost algorithm.

G. XGBoost Algorithm

XG Boost stands for eXtreme Gradient Boosting. This algorithm is known for its speed, computational efficiency and performance, even for large datasets [24], [25]. It is similar to gradient boosting apart from the fact that it performs regularization additionally in the objective function. A gradient descent procedure is used to minimize the loss of the function similar to the gradient boosting algorithm.

VII. Results

Table.1 Tabulation of accuracy with respect to each traditional comparison algorithm.

Algorithm	**Accuracy(in %)**
Sequence Matcher	**65.29**
Cosine Similarity	64.11

A. Traditional methods

The two traditional comparison techniques, Sequence matcher and Cosine similarity gave an accuracy value of 65.29 and 64.11 respectively as shown. The values were tabulated (see Table.1).

B. Machine learning approaches

Table.2 Tabulation of accuracy with respect to each Machine learning algorithm after applying Count-Vectorizer.

Algorithm	**Accuracy(in %)**
Naïve bayes	73.39
Logistic Regression	75.45
Shallow neural network(ANN)	77.93
XGB	**80.89**
SVM	69.37
RNN	79.63

After the completion of pre processing of data, the feature extraction step is done in 2 ways-one using Count-Vectorizer and the other using TFIDF-Vectorizer, both of them being Bag of Words models. After that the ML techniques were applied for modeling and the results are tabulated (see Table.2). Logistic Regression gave an accuracy score of value 75.45. Naïve Bayes classifier gave an accuracy score of 73.39. The XGB gave the highest accuracy among the models taken into consideration, with a score of 80.89. RNN (with LSTM) model gave an accuracy of 78.90. The shallow neural networks gave an accuracy value of 77.93. The SVM, with RBF kernel, gave the least value of 69.37, which was the lowest of the lot.

Table.3 Tabulation of accuracy with respect to each Machine learning algorithm after applying TFIDF-Vectorizer.

Algorithm	**Accuracy(in %)**
Naïve bayes	71.43
Logistic Regression	74.38
Shallow neural network(ANN)	76.84
XGB	**80.12**
SVM	68.76
RNN	78.94

The second case was to extract the features using TFIDF-Vectorizer. After that the ML techniques were applied for modeling and the results were tabulated (see Table.3). For Logistic Regression, it gave an accuracy score of value 74.38. Then it was modeled with Naïve Bayes classifier which gave an accuracy score of 71.43. The XGB model gave the highest accuracy in the second case also, with a score of 80.12. RNN (with LSTM) model gave an accuracy of 78.94. The Shallow neural networks gave an accuracy value of 76.84. The SVM, with RBF kernel, gave the least value of 68.76.

The results suggest that the dataset works efficiently when applied with Count-Vectorizer as suggested by the performance of each model with both the BOW models. The XGB model gave the best performance in both the cases but the XGB model with Count-Vectorizer out-performed the XGB model with TFIDF-Vectorizer by a slight margin. Another important fact enlightened by the results is that Machine Learning models predicted the similarity better than the traditional methods, which gave comparatively very less accuracies while predicting the similarity.

VIII. CONCLUSION

In this paper, the Quora Question Pairs dataset is used to predict the similarity between two questions for duplicity using both Non-Machine Learning methods and Machine Learning methods. Among the traditional methods, Sequence matching algorithm gave the best accuracy of 65.29% after applying the basic NLP techniques. The dataset was pre processed in 2 different ways before applying the ML algorithms, one using Count-Vectorizer and other using TFIDF-Vectorizer. In both the cases, XGB gave the highest accuracy of 80.89% for Count-Vectorizer and 80.12% for TFIDF-Vectorizer. All the Machine Learning algorithms taken into considerations performed better than the Non-ML algorithms suggesting ML algorithms are efficient in predicting the similarity between the 2 questions. The models gave a comparable performance with Siamese networks while also achieving computational efficiency at the same time. In the future, word embeddings like GLoVe and Word2Vec can be used instead of BOW models, Count-Vectorizer and TFIDF-Vectorizer, to get deeper contextual information from the questions, which can improve the efficiency of the models especially the RNN-LSTM model.

REFERENCES

1. Rahutomo, Faisal & Kitasuka, Teruaki & Aritsugi, Masayoshi, (2012), “Semantic Cosine Similarity”.
2. Gunawan, Dani & A Sembiring, C & Budiman, Mohammad. (2018), “The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents”, Journal of Physics: Conference Series. 978. 012120. 10.1088/1742-6596/978/1/012120.
3. Ojstersek, Milan & Ferme, Marko. (2011), “Text analysis with sequence matching. International journal of computers”, 5. 235-242.
4. P. W. Adriaans, D. Zantinge, Data Mining, Addison-Wesley, 1996.
5. S. Agarwal, "Data Mining: Data Mining Concepts and Techniques," *2013 International Conference on Machine Intelligence and Research Advancement*, Katra, 2013, pp. 203-207.
6. Ikonomakis, Emmanouil & Kotsiantis, Sotiris & Tampakas, V. (2005), “Text Classification Using Machine Learning Techniques”, WSEAS transactions on computers, 4. 966-974.
7. Zihan Chen, Hongbo Zhang, Xiaoji Zhang, Leqi Zhao, “Quora Question Pairs”
8. Homma, Yukiko, “Detecting Duplicate Questions with Deep Learning.”, (2017).
9. Zhang, Yin & Jin, Rong & Zhou, Zhi-Hua, (2010), “Understanding bag-of-words model: A statistical framework”, International Journal of Machine Learning and Cybernetics, 1. 43-52. 10.1007/s13042-010-0001-0.
10. Kaviani, Pouria & Dhotre, Sunita, (2017), “Short Survey on Naive Bayes Algorithm”, International Journal of Advance Research in Computer Science and Management. 04.
11. Wang, Yong & Hodges, Julia & Tang, Bo, (2003), “Classification of Web Documents Using a Naive Bayes Method”, IEEE Transactions on Applications and Industry, 560- 564. 10.1109/TAI.2003.1250241.
12. Basheer, Imad & Hajmeer, M.N.. (2001), “Artificial Neural Networks: Fundamentals, Computing, Design, and Application”, Journal of microbiological methods, 43. 3-31. 10.1016/S0167-7012(00)00201-3.
13. Zhang, Zhongheng, (2016), “A gentle introduction to artificial neural networks. Annals of Translational Medicine”, 4. 370-370. 10.21037/atm.2016.06.20.
14. Shafi, Imran & Jamil, Ahmad & Shah, Syed & M Kashif, Faisal, (2007), “Impact of Varying Neurons and Hidden Layers in Neural Network Architecture for a Time Frequency Application”, 188 - 193. 10.1109/INMIC.2006.358160.
15. Bodén, Mikael, (2001), “A Guide to Recurrent Neural Networks and Backpropagation”.
16. Du, K.-L & Swamy, M.N.s, (2014), “Recurrent Neural Networks”, 10.1007/978-1-4471-5571-3_11.
17. Hochreiter, Sepp & Schmidhuber, Jürgen, (1997), “Long Short-term Memory. Neural computation”, 9. 1735-80. 10.1162/neco.1997.9.8.1735
18. Joanne Peng, Kuk Lida Lee, & Gary M. Ingersoll, “An Introduction to Logistic Regression Analysis and Reporting”, Journal of Educational Research - J EDUC RES. 96, 3-14. 10.1080/00220670209598786, 2002
19. Miftar Ramosacaj, Vjollca Hasani & Alba Dumi, “Application of Logistic Regression in the Study of Students’ Performance Level (Case Study of Vlora University)” Journal of Educational and Social Research. 10.5901/jesr.2015.v5n3p239, 2015
20. M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines,” in *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18-28, July-Aug. 1998.
21. Kwang In Kim, Keechul Jung, Se Hyun Park and Hang Joon Kim, "Support vector machines for texture classification," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1542-1550, Nov. 2002.
22. Yekkehkhany, Bahareh & Safari, Abdolreza & Homayouni, Saeid & Hasanlou, Mahdi, (2014), “A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data”, ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-2/W3. 281-285. 10.5194/isprsarchives-XL-2-W3-281-2014.
23. Bühlmann, Peter, “Bagging, Boosting and Ensemble Methods”, Handbook of Computational Statistics. 10.1007/978-3-642-21551-3_33, 2012
24. Santhanam, Ramraj & Uzir, Nishant & Raman, Sunil & Banerjee, Shatadeep, (2017), “Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets”.
25. Tianqi Chen & Carlos Guestrin, “XGBoost: A Scalable Tree Boosting System”. 785-794. 10.1145/2939672.2939785, 2016

AUTHORS PROFILE

Rishickesh Ramesh is working toward B.Tech degree in the Department of Information Technology, SSN College of Engineering. His research interests include machine learning techniques for natural language processing and understanding, data analytics, Internet of Things and deep learning.

R.P Ram Kumar is working toward B.Tech degree in the Department of Information Technology, SSN College of Engineering. His research interests include machine learning techniques for natural language processing and understanding, data analytics and deep learning

A. Shahina is a professor in the department of Information Technology at SSN. She has 14 years of teaching and research experience, with over 5 years of research exclusively in the field of Speech Processing, one of the widely growing and popular research areas. She aims to develop speech based clinical applications, and technologies for viable biometric person authentication systems through sustained research. Her areas of interest include machine learning, deep learning, and speech processing.

Dr. A. Nayeemulla Khan is the Dean Academics and Professor in the School of Computing Science and Engineering at VIT Chennai. He was previously associated with the Airports Authority of India as a Senior Manager ATC and as a Research Scientist at Acusis Software India Pvt. Ltd. His areas of interest include speech and speaker recognition, machine learning, brain computer interface among others.