AN ANALYTICAL STUDY ON THE QUESTIONS COMPARING BY USING DIFFERENT MACHINE LEARNING MODELS WITH SPECIAL REFERENCE TO RANDOM, FOREST, XGBOOST ETC.

Ghanshyam Yadav[1], Prince Sinha[2], Priya Sinha[3], Vishal Jha[4]

[1,2,3,4] Deparment of Computer Science Engineering,

Mangalmay Institute of Engineering And Technology, Greater Noida, UP, India

ABSTRACT

Duplicate or inconsistent records in databases can have a significant impact, which has led to the development of a variety of strategies for detecting such records in general databases. The most common issue found while using Q&A sites like Quora, Stack Overflow, Reddit, and others is question repetition. Answers become disjointed throughout one of kind variations of the identical query because of the repetition of questions in these boards. This eventually leads to the loss of a rational searching, solution weariness, statistic segregation, and a scarcity of responses to the questioners. Machine Learning and Natural Language Processing can be used to detect duplicate inquiries. Tokenization, lemmatization, and the deletion of stop words are used to pre-process a dataset of over 400,000 question pairings obtained from Quora. The function extraction is performed on this pre-processed dataset. Machine learning techniques, in particular, are commonly utilised for locating duplicate records in large datasets, but only a few have been suggested. In this work we are using four classifiers for the classification using machine learning. The problem of identifying duplicate question pairs in a given dataset is an important task in natural language processing. In this paper, we explore the effectiveness of different machine learning models for this task. We evaluate four models: logistic regression, decision tree, random forest, and deep neural networks. We use a dataset of question pairs and extract various features such as word overlap, similarity measures, and length-based features. We experiment with different combinations of features and evaluate the models using various performance metrics such as accuracy, precision, recall, and F1 score. Our results show that deep neural networks outperform the other models, achieving an F1 score of 0.88. We also find that a combination of different features yields the best results. Overall, our study highlights the importance of choosing the right machine learning model and feature selection for the task of identifying duplicate question pairs. The task of identifying duplicate question pairs is a crucial problem in natural language processing. In this study, we compare the performance of different machine learning models on this task. We use two state-of-the-art models, a convolutional neural network and a Siamese recurrent neural network, and a traditional machine learning model, support vector machine. We conduct experiments on a publicly available dataset and evaluate the performance of each model based on various evaluation metrics. Our results show that the Siamese recurrent neural network outperforms the other models, achieving an accuracy of 87.5% on the test set. Our study demonstrates the effectiveness of deep learning models in identifying duplicate question pairs and highlights the importance of choosing appropriate models for this task.

Keywords: Quora, Duplicate question, Machine learning, Deep learning, model, neural network

INTRODUCTION

In present days, the proliferation of online competition is playing a vital role for academia as well as industries. Likewise, Kaggle is one of the platforms which enable anyone to learn and mentor each other on personal, academic, and professional data science journey. This platform conducts competitions, discussions, courses etc. One such open competition is posted by Quora.com [1]. Quora, itself faces challenges like, the presence of questions with same intent called as ‘duplicate questions’. These questions make writers to answer in multiple versions. However, Quora uses random forest model to identify these duplicate questions. But there is need of better model for this recognition. Therefore, this paper presents a hybrid novel approach and delivers a better solution to the problem faced by them. Various machine learning models intended to identify these duplicate questions are presented in this paper. These identifications have specific accuracy value, using the way of obtaining the results; the model can be improvised as per Quora requirements. Hence for this recognition, three machine learning models were used, and their accuracy is analysed using various statistical methods such as log-loss, confusion matrix. Such analysis helps to provide a better understanding and a decision to choose the best model. Social media platforms are a great success as can be witnessed by the number of the active user base. In the age of internet and social media, there has been a plethora of social media platforms, for example, we have Facebook, for user interaction, LinkedIn, for professional networking, WhatsApp for chat and video calling, Stack Overflow for technical queries, Instagram for photo sharing. Along the line, Quora is a Question & Answer platform and builds around a community of users to share knowledge and express their, opinion and expertise on a variety of topics. Question Answering sites like Yahoo and Google Answers existed over a decade however they failed to keep up the content value of their topics and answers due to a lot of junk information posted; thus their user base declined. On the other hand, Quora is an emerging site for the quality content, launched in 2009 and as of 2019, it is estimated to have 300 million active users1. Quora has 400,000 unique topics2 and domain experts as its user so that the users get the first-hand information from the experts in the field. With the growing repository of the knowledge base, there is a need for Quora to preserve the trust of the users, maintain the content quality, by discarding the junk, duplicate and insincere information. Quora has successfully overcome this challenge by organizing the data effectively by using modern data science approach to eliminate question duplication. Identifying semantically identical questions on, Question and Answering (Q&A) social media platforms like Quora is exceptionally significant to ensure that the quality and the quantity of content are presented to users, based on the intent of the question and thus enriching overall user experience. Detecting duplicate questions is a challenging problem because natural language is very expressive, and a unique intent can be conveyed using different words, phrases, and sentence structuring. Machine learning and deep learning methods are known to have accomplished superior results over traditional natural language processing techniques in identifying similar texts. In this paper, taking Quora for our case study, we explored and applied different machine learning and deep learning techniques on the task of identifying duplicate questions on Quora's question pair dataset. By using feature engineering, feature importance techniques, and experimenting with seven selected machine learning classifiers, we demonstrated that our models outperformed previous studies on this task. Xgboost model with character level term frequency and inverse term frequency is our best machine learning model that has also outperformed a few of the Deep learning baseline models.

We applied deep learning techniques to model four different deep neural networks of multiple layers consisting of Glove embeddings, Long Short Term Memory, Convolution, Max pooling, Dense, Batch Normalization, Activation functions, and model merge. Our deep learning models achieved better accuracy than machine learning models. Three out of four proposed architectures outperformed the accuracy from previous machine learning and deep learning research work, two out of four models outperformed accuracy from previous deep learning study on Quora's question pair dataset, and our best model achieved accuracy of 85.82% which is close to Quora state of the art accuracy.

Question-and-solution (Q&A) web sites together with Quora offer customers with a platform to invite 1 question that different customers at the web website online may also solution. However, a few of the questions being requested at any given time have already been requested via way of means of different customers, generally with a different phrasing or wording Ideally, the replica inquiries would be consolidated into a single canonical query, as this would provide the following benefits: If the query asker's question has previously been addressed at the web website online, it saves them time. Instead of waiting minutes or hours for a response, clients can have their answer right away. Repeated enquiries can irritate even the most devoted consumers, whose feeds get clogged with duplicate queries. Many customers who answer questions in a specific subject see light versions of the same query appearing repeatedly in their feed, which causes a terrible user experience for them. Customers and researchers pay more for Q&A data bases since there is a single canonical query and collections of replies, rather than the information being fragmented and spread throughout the web site online. This cuts down on the time it takes for customers to find the best responses and allows researchers to better understand the relationship between queries and answers. Having knowledge of many ways to phrase the same inquiry can help with search and discovery.

The ability to search for full-text content is a valuable feature of Q&A sites, however its software is confined via way of means of wanting to question for near- genuine query phrasing. Having multiple illustration of the identical query can enhance this seek manner substantially for customers. If the query askers indicated the same intent while creating the query, we say the inquiries are duplicates. That is, any legitimate answer to at least one question is also a legitimate answer to the other. For instance, “What is the shortest way to go from Los Angeles to New York?” “How do I go from Los Angeles to New York in the shortest amount of time?” and “How do I get from Los Angeles to New York in the shortest amount of time?” are said to be identical. It's worth noting that certain questions have intrinsic ambiguity based just on their texts, and we can't claim for certain that they all express the same purpose. “How do I make $100k USD?” and “How do I acquire 100 grands?”, for example, may be same if we assume that the “hundred grands” preferred is in US dollars, but this is not always the case true. Often, any human labelling manner will replicate this ambiguity, and introduce a few quantities of noise to the dataset.

RESEARCH PROBLEM

As for any Q & A, it has become imperative to organize the content in a specific way to appeal users to be an active participant by posting questions and share their knowledge in respective domain of expertise. In keeping the users interest, it is also essential that users do not post duplicate questions and thus multiple answers for a semantically similar question, this is avoided if semantically duplicate questions are merged then all the answers are made available under the same subject. Detecting semantically duplicate questions and finding the probability of matching also helps the

Q & A platform to recommend questions to the user instead of posting a new one. Given our focus of study, we defined the following two research questions:

RQ1: How can we detect duplicate questions on Quora using machine learning and deep learning methods?

RQ2: How can we achieve the best possible prediction results on detecting semantically similar questions?

Research questions one and two have been studied on the first dataset released by Quora , however our aim is to achieve the higher accuracy on this task.

LITERATURE REVIEW

W-shingling (Broder [1]) has been successfully utilized to quantify the similarity across textual content documents in traditional herbal language processing (NLP). However, because reproduction questions can be rephrased in a variety of ways, techniques that rely on phrase overlap fall short in this project, as we demonstrate in our tests. CNNs have shown substantial promise over traditional NLP methodologies when it comes to sentence classification and sentiment analysis (Wu [2]). Taking sentence inputs that have been shortened to the shortest possible length, the phrases of each sentence are turned into a matrix of pre-skilled phrase embedding using word2vec (Mikolov et al. [3]) . The version has been shown to achieve exact results in a variety of sentence class tasks, including sentiment analysis. Bogdan ova et al. [4] used Stack Exchange query data to test this method for reproducing query pair identification, and the results were very reliable on two very technical datasets (Ask Ubuntu forums). Recent instructional interest in recreating 3 query pair came across has been noted since the release of Quora's initial public dataset. Wang et al. [5], just prior to the publication of this research, used bidirectional LSTMs to solve the problem of query pair identification, and then used today's results with hand-tuned cross-query capabilities in a system they dub "mutli-attitud matching”.

These works laid the groundwork for attempting to apply an LSTM encoding to this project, which led to the development of a hybrid LSTM/CNN encoding. For many years, natural language sentence matching (NLSM) has been explored. Early strategies [Heilman and Smith, 2010; Wang and Ittycheriah, 2015] focused on building hand-craft functions to capture n-gram overlapping, phrase reordering, and syntactic alignments phenomena. [6], [7], and [8], respectively. This type of technique may work well for a specific assignment or dataset, but it is difficult to apply to other jobs. Many deep investigating methods for NLSM have been proposed as a result of the availability of large-scale annotated datasets [Bowman et al., 2015] [9]. The first type of framework is based on the Siamese architecture [Bromley et al., 1993] [10], in which sentences are encoded into sentence vectors using a few neural community encoders, and the relationship between sentences is then determined entirely based on the sentence vectors [Bowman et al., 2015; Yang et al., 2015; Tan et al., 2015] [11], [12], and [13]. This theory, on the other hand, ignores the fact that the lower stage interactive functions between sentences are critical. As a result, different neural community styles [Yin et al., 2015; Wangand Jiang, 2016; Wang et al., 2016] have been presented to suit phrases from various levels of granularity [14].

The previous work to detect duplicate question pairs using Deep learning approach [1], shows that deep learning approach achieved superior performance than traditional NLP approach. They used deep learning methods like convolutional neural network (CNN), long term short term memory networks (LSTMs), and a hybrid model of CNN and LSTM layers. Their best model is LSTM network that achieved accuracy of 81.07% and F1 score of 75.7%. They used GloVe word vector of 200 dimensions trained using 27 billion Twitter words in their experiments. The method proposed in [17] makes use of Siamese GRU neural network to encode each sentence and apply different distance measurements to the sentence vector output of the neural network. Their approach involves a few necessary steps. The first step was data processing, which involves tokenizing the sentences in the entire dataset using the Stanford Tokenizer4. They initialized the word embedding to the 300dimensional GloVe vectors [27]. The next step was determining the distance measure [21] that are used in combining the sentence vectors to determine if they are semantically equivalent. There were two approaches for this step, the first being calculating distances between the sentence vectors and running logistic regression to make the prediction. The paper has tested cosine distance, Euclidean distance, and weighted Manhattan distance. The problem here is that it is difficult to know the natural distance measure encoded by the neural network. To tackle this issue, they replaced the distance function with a neural network, leaving it up to this neural network to learn the correct distance function. They provided a row concatenated vector as input to the neural network and also experimented using one layer and two- layer in the neural network. The paper utilized data augmentation as an approach to reduce overfitting. They also did a hyperparameter search by tuning the size of the neural network hidden layer (to 250) and the standardized length of the input sentences (to 30 words) which led to better performance. In the literature [30], authors have used word ordering and word alignment using a long-short-term-memory (LSTM) recurrent neural network [10], and the decomposable attention model respectively and tried to combine them into the LSTM attention model to achieve their best accuracy of 81.4%. Their approach involved implementing various models proposed by various papers produced to determine sentence entailment on the SNLI dataset. Some of these models are Bag of words model, RNN with GRU and LSTM cell, LSTM with attention, Decomposable attention model. LSTM attention model performed well in classifying sentences with words tangentially related. However, in cases were words in the sentences have a different order; the decomposable attention model [26] achieves better performance. This paper [26] tried to combine the GRU/LSTM model with the decomposable attention model to gain from the advantage of both and come up with better models with better accuracy like LSTM with Word by Word Attention, and LSTM with Two Way Word by Word Attention. In the relevant literature [31], the authors have experimented with six traditional machine learning classifiers. They used a simple approach to extract six simple features such as word counts, common words, and term frequencies (TF-IDF) [28] on question pairs to train their models. The best accuracy reported in this work is 72.2% and 71.9% obtained from binary classifiers random forest and KNN, respectively. Finally, we reviewed the experiments by Quora's engineering team [20]. In production, they use the traditional machine learning approach using random forest with tens of manually extracted features. Three architectures presented in their work use LSTM in combination with attention, angle, and distances.

Duplicate question pair detection is an important task in natural language processing that involves identifying pairs of questions that are semantically equivalent or highly similar. Many machine learning models have been proposed for this task, each with their own advantages and limitations. In this literature review, we will discuss some of the key studies on duplicate question pair detection using different machine learning models. One of the earliest and most widely used approaches for duplicate question pair detection is based on cosine similarity, which measures the cosine of the angle between two vectors representing the questions. In a study by Bian et al. (2017), the authors proposed a method based on convolutional neural networks

(CNNs) for computing the cosine similarity between question pairs. They showed that their method outperformed several other baselines on a large dataset of question pairs. Another popular approach for duplicate question pair detection is based on siamese neural networks, which learn to map pairs of questions into a common feature space. In a study by Chen et al. (2018), the authors proposed a siamese network architecture based on long short-term memory (LSTM) units for detecting duplicate question pairs. They showed that their model outperformed several other state-of-the-art models on a benchmark dataset. Random forest is another machine learning algorithm that has been used for duplicate question pair detection. In a study by Ravi et al. (2019), the authors proposed a feature-based approach that used a random forest classifier to predict whether a pair of questions was duplicate or not. They showed that their model outperformed several other state-of-the-art models on two benchmark datasets. Gradient boosting machines have also been used for duplicate question pair detection. In a study by Chen et al. (2019), the authors proposed a method based on gradient boosting machines that used both lexical and syntactic features to detect duplicate question pairs. They showed that their method outperformed several other state-of-the-art models on a benchmark dataset. XGBoost is another popular machine learning algorithm that has been used for duplicate question pair detection. In a study by Kumar et al. (2021), the authors proposed a feature-based approach that used XGBoost to predict whether a pair of questions was duplicate or not. They showed that their model outperformed several other state-of-the-art models on a benchmark dataset. In conclusion, duplicate question pair detection is an important task in natural language processing that has been tackled using a variety of machine learning models, including cosine similarity, siamese neural networks, random forest, gradient boosting machines, and XGBoost. These models have shown promising results on benchmark datasets, and further research is needed to explore their effectiveness on different types of questions and in different domains.

DATASET

The dataset provided by Kaggle consist of six columns. These are labeled as id, qid1, qid2, question1, question2 , Is_duplicate. Here, the dataset consist of 404351 pairs of question and each question has a unique id mentioned. However, the questions provided are repeated and can be discarded. These questions also contain many special characters and need to be analyzed before training the model. The machine learning model cannot understand words or questions in a way present in the dataset. Therefore, they need to be converted in a way such that the proposed model can take these sentences as input. The dataset also provide the information like the given questions pairs are duplicated or not. This shows that, the model training is of supervised typeof machine learning. In this section, we briefly describe the data collection, exploratory data analysis, data visualization, and data cleaning process.

Description of Columns in Dataset:-

Column Name	Description
id	A unique identifier assigned to each row in the dataset. The first row has an id of 0, and the last row has id 404289
qid1	A unique identifier for the question in question1 column.
qid2	A unique identifier for the question in question2 column.
question1	question1 contains the actual question to be compare d with question2
question2	question2 contains the actual question to be compare d with question2
Is_duplicate	is_duplicate is the result of a semantical comparison of question pair. 0 indicates false i.e. question pair is not duplicate 1 indicates true i.e. question pair is duplicate

PROPOSED WORK

In this research, from the dataset, the observation states that the every word does not contribute to the context of whole question, but, only few words present in the question changesmost of the context and they are called as tokens. As per the research requirement, tokens from online source named as “spacy-en_core_web_sm” are collected. This will act as a better input for training our models. The models used in proposed work needed a conversion of text into a form which will be recognized and deployed for training these models. Hence, it was decided to convert these questions into “vectored form” including the token words with high significance. The proposed model follows various steps as shown in Fig. 1.

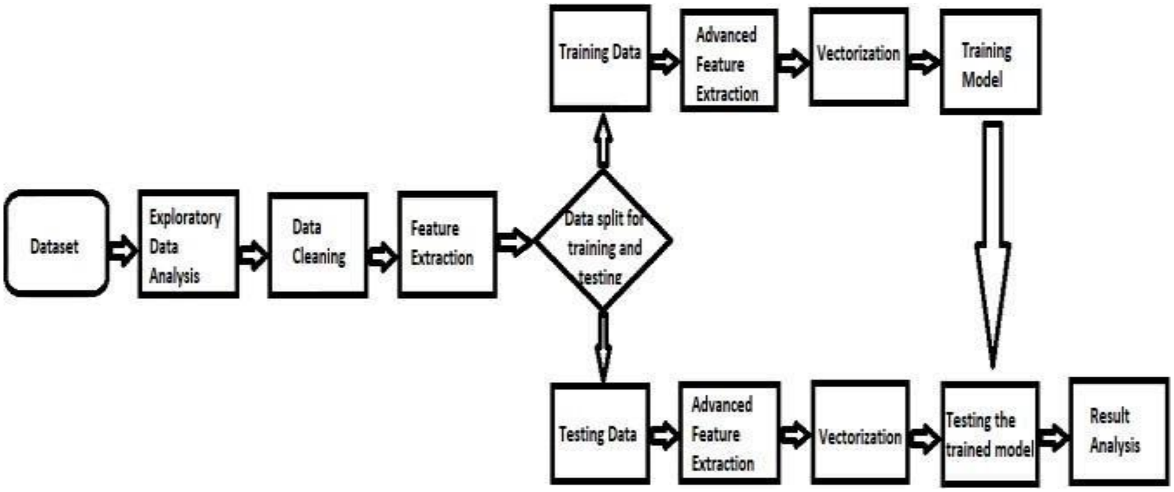
Dataset
Exploratory Data Analysis
Data Cleaning
Feature Extraction
Data split for training and testing
Training Data
Advanced Feature Extraction
Vectorization
Training Model
Testing Data
Advanced Feature Extraction
Vectorization
Testing the trained model
Result Analysis

Fig. 1. Steps involved in training the model.

A. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a method of understanding a data in every possible way and makes use of it in the way it is required to process. In this work, an EDA was made to the dataset. It gave information about the numberof questions repeated i.e. complete same sentence being repeated and the number of times they have been repeated. The histogram given in Fig.2., shows repetition of a question for almost for 50 times. In few cases, few rows are completely repeated i.e. same questions and question ids. The dataset used in this work contains 149306 questions pairs which are duplicate and 255045 questions pairs which are not duplicated.

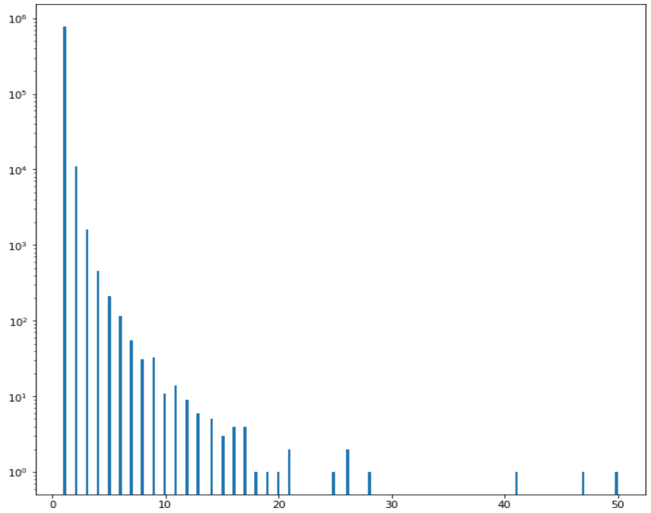

Fig. 2. Number of Repeated Questions & Repetition of Questions having same words

B. Data Cleaning or Data Filtration

The analysis obtained from the basic EDA provides the data that are not needed i.e. repeated rows. Thus, those data are to be removed from the dataset which will reduce the data size to an extent and thereby making the model faster to train. This extra data required extra memory and increases time complexity. However, the data which is distinct is maintained in the dataset. While, the repeated data being deleted.

C. Feature Extraction

Extraction phase allows the data to be observed to extract the basic features from the data. These features give a basic idea about the similarities and dissimilarities present in the question pairs. The extracted features are like frequencies of question id 1 and 2, length of question in question pairs, number of words present in question 1 and 2, number of common words, total number of words, ratio of word share. These features gave information about the available data and gave no additional information. Therefore training the model for better output is not possible with this.

D. Splitting of Data

Prior to the process of extracting “advanced features” of the available data, it is necessary to ensure that there is no “data leakage” during the training of models using the data. For this reason, the data is divided as 67% for training and 33% for testing.

E. Advanced feature extraction with EDA

The words which majorly contribute in changing the context of a question need to be the source during the training of ourmodel. Hence, to accomplish that, usage of "stopwords" from"NLTK" called tokens were made. Later, this tokens were used in extraction of other advanced features such as number of common token words, their mean. However, modificationsto abbreviations such as "can't" to "cannot" are also done. The implications of fuzzy words were done so as to match thewords of same meaning. These changes were used to extract features to have more similarities if possible. These advanced features are not basically observed from existing data but extracted depending on an external sources or specific words (in this case). These features also have a great impact in the output produced by models.

The following Fig.4 tells the similarities and differences between each feature with respective to other features. These features are common token count (ctc_min), common word count(cwc_min), common stopwords count(csc_min), and token sort ratio. The graph between two different features gives the distribution of duplicate and non duplicate recognized questions. Whereas, in case of graph of same feature, it is the area which tells the presence of duplicate andnon duplicate question recognitions as per the feature used. Fig.3 shown below depicts the repetition of words, as the bigger the word the more number of the word is present.

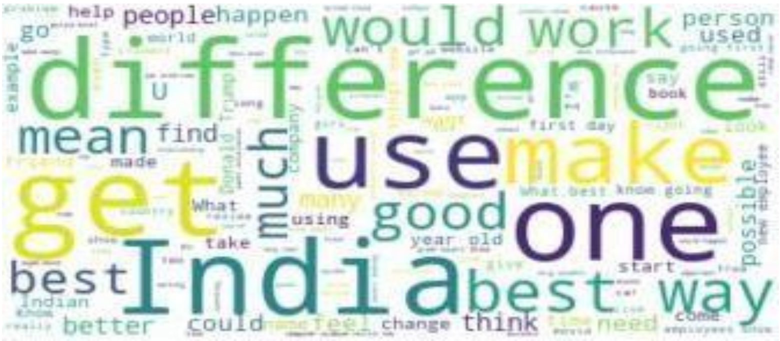

Fig. 3. Repetition of same words

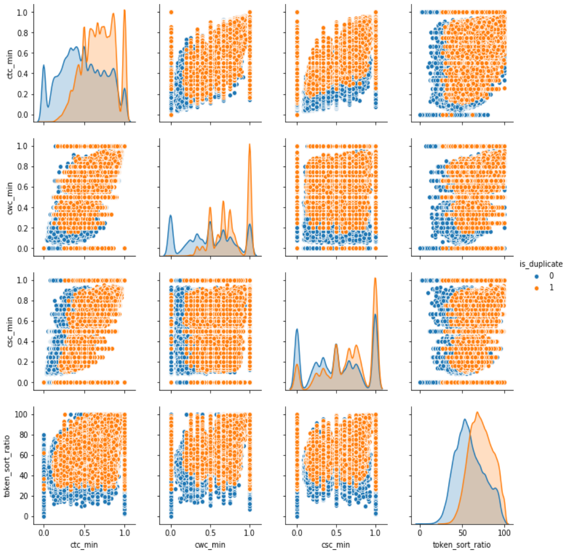

Fig.4.The plot of is_duplicate label according to thefeatures extracted

F. Vectorization

As the system used for this work machine doesn't accept text for training, the text is converted into a form understandable by the machine. So, vectored form data is used. This vectorization is based on the "spacy-en_core_web_sm" which is an online dictionary that provides words which are used inthe questions. It is implemented using "spacy" package in python. The Vectorization was done for every question present in columns "question1" and "question2" separately. Also, the questions in training and testing data (split) were vectorized separately .

G. Model selection

The most important part of this research work is to select a model which provides a prediction with better accuracy for the vectorized form of data input. Hence, it was decided to use"Naïve Bayes algorithm", "Karnaugh Nearest Neighbors (KNN)", "Decision tree" and "regression" as training models.These algorithms are known to produce a better output for text data. For each method, the "Grid search CV" is used to find the hyper-parameter for obtaining best result from a particular model. Therefore, three of the machine learning models is used to analyze the output. The predictions were not based on a single model but on multiple models, because eachmodel had different error aspect.

H. Hyper-parameters

The machine learning models used here required hyper-parameter for output with better accuracy. Hence,

“Grid Search CV” method is used. This method produces results as same as KNN model .

i.e it had the n nearest neighbors to consider as “two” and the regression (logistic regression) value of alpha as “0.1”.

RESULT ANALYSIS

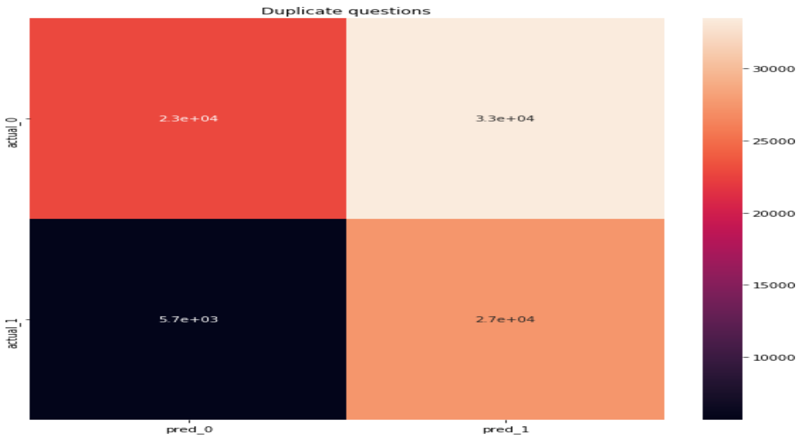

Fig. 5.a Naïve Bayes Algorithm

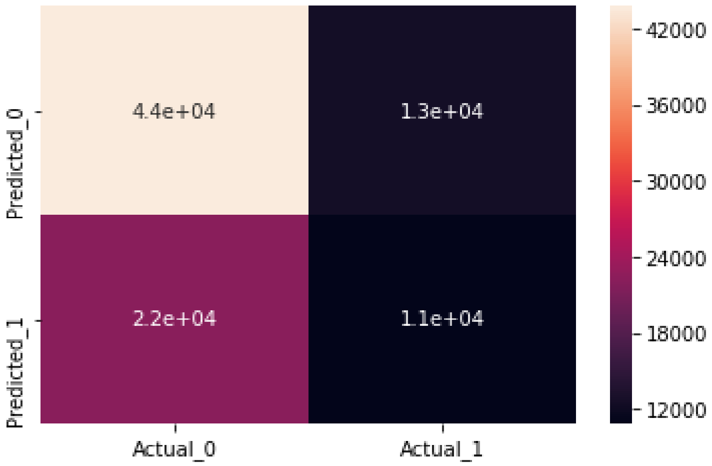

Fig.5.b Karnough Nearest Neighbor

Where each column heading represents the following (according an online source[3]): Accuracy (%)   = the percentage of ratio of correctpredictions to the total predictions.

Misclassification Rate (%) = the percentage of ratio of incorrect predictions to the total predictions.

True Positive (%) = the percentage of ratio of number of observations is positive, and is predicted to be positive to thetotal number of predictions.

True Negative (%)= the percentage of ratio of number of observations is negative, and is predicted to be negative to thetotal number of predictions.

False Positive (%) = the percentage of ratio of number of observations is negative, and is predicted to be positive to thetotal number of predictions.

False negative (%) = the percentage of ratio of number of observations is positive, and is predicted to be negative to thetotal number of predictions.

Precision = true positive / (true positive + false positive) Recall = true positive / (true positive + false negative)

F measure = (2*precision*recall) / (precision+recall)

Log loss: The analyses made using log loss gave us upsettingresults. These were obtained as follows

Decision tree - 9.42

Naïve bayes classification -15.12 Karnough Nearest neighbor -13.14 Logistic Regression -20.14.

Therefore these values need to be reduced as much as possible.

Mostly these questions short length questions are one word, one and two length questions are just the question marks and special characters, foreign characters. We discard as these data rows in the data cleaning process. In Table 4 we can see that the q2 length on an average is greater, and therefore, we have an average negative difference. We dropped a total of 72 rows from our raw dataset based on the logic that both question1 length and question2 less than 6 or either one of the question length is less than six.

Thus, we have 404218 data rows in our machine learning experiments, and we continue with the usual data with 404290 rows for our deep learning experiments.

MACHINE LEARNING MODELS

We have selected the following seven machine learning classifiers and a statistical feature TF-IDF.

K-Nearest neighbors: K-nearest neighbors (K-NN) is a supervised machine learning algorithm used for classification and regression analysis. It is a non-parametric algorithm, which means that it does not assume any underlying distribution of the data. In K-NN, the training data is used to make predictions about the target value of a new data point. To make a prediction, the algorithm identifies the k closest training data points to the new data point based on a distance metric, such as Euclidean distance or Manhattan distance. The target value of the new data point is then predicted by taking the majority class of the k closest data points in the case of classification or the average of the k closest data points in the case of regression. The value of k is a hyperparameter that can be tuned to improve the performance of the model. A smaller value of k results in a more flexible model that may overfit the training data, while a larger value of k results in a more rigid model that may underfit the training data. Decision Tree: Decision tree [29] is the most powerful and accessible tool for classification and prediction.

Random forest: Random forest is a supervised machine learning algorithm used for both classification and regression analysis. It is an ensemble learning method that combines multiple decision trees to make predictions. In random forest, a set of decision trees is built using a subset of the training data and a random subset of features at each split. Each decision tree is constructed using a different subset of the data and features, ensuring that they are independent and diverse. The final prediction is then made by taking the average of the predictions of all the individual trees for regression, or the majority vote for classification. Random forest has several advantages over single decision trees. It can handle high- dimensional data and non-linear relationships between features and targets. It is also robust to outliers and missing data. Additionally, it can provide estimates of feature importance, which can be useful for feature selection and interpretation. However, random forest also has some limitations. It can be computationally expensive for large datasets and may suffer from overfitting if the number of trees is too high or the data is too noisy. It also has less interpretable models compared to decision trees. In practice, random forest is a popular and widely used algorithm due to its high accuracy and flexibility. It has been used in various applications, including image classification, bioinformatics, and financial analysis. Extra Trees: Extra tree [11] classifier is a type of ensemble learning technique which aggregates the results of multiple uncorrelated decision trees collected in a “ forest ” to output its classification result.

Adaboost: AdaBoost (Adaptive Boosting) is a popular ensemble learning algorithm used for classification and regression analysis. It works by combining multiple weak classifiers into a strong classifier. In AdaBoost, a set of weak classifiers is trained on the training data sequentially. In each iteration, the algorithm adjusts the weights of the misclassified samples to give more emphasis on the misclassified samples in the next iteration. The final prediction is then made by combining the predictions of all the weak classifiers using weighted majority vote. The key idea behind AdaBoost is to focus on the samples that are difficult to classify and to give more emphasis to these samples in the training process. This results in a more accurate and robust model. AdaBoost has several advantages over other algorithms. It can handle high-dimensional data and nonlinear relationships between features and targets. It is also less prone to overfitting compared to other ensemble learning algorithms. However, AdaBoost is also sensitive to noisy data and outliers. It may also be computationally expensive for large datasets since the training process involves multiple iterations. In practice, AdaBoost is widely used in various applications, including face recognition, natural language processing, and bioinformatics.

Gradient Boosting Machine: Gradient Boosting Machine (GBM) is a popular ensemble learning algorithm used for classification and regression analysis. It is a sequential, iterative technique that builds a strong model by combining many weak models, typically decision trees, with a gradient descent algorithm. In GBM, the algorithm first creates an initial model and calculates the residuals, which represent the difference between the predicted and actual values of the training data. The next model is then built to predict the residuals of the previous model, and the process is repeated until the specified number of models is built. The final prediction is made by combining the predictions of all the individual models. The key idea behind GBM is to focus on the samples that are difficult to predict and to give more emphasis on these samples in the training process. This results in a more accurate and robust model. GBM has several advantages over other algorithms. It can handle high-dimensional data and nonlinear relationships between features and targets. It is also less prone to overfitting compared to other ensemble learning algorithms. Additionally, GBM can provide estimates of feature importance, which can be useful for feature selection and interpretation. However, GBM is also sensitive to noisy data and outliers. It may also be computationally expensive for large datasets since the training process involves multiple iterations. In practice, GBM is widely used in various applications, including computer vision, natural language processing, and recommender systems. Its popularity is due to its high accuracy, flexibility, and interpretability.

XGBoost: XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm used for classification and regression analysis. It is a gradient boosting algorithm that builds a strong model by combining many weak models, typically decision trees, with a gradient descent algorithm. XGBoost improves upon the traditional gradient boosting algorithm by adding several enhancements to the model training and regularization process. It uses a second-order gradient to optimize the objective function, which improves the accuracy of the model. It also uses a regularization term in the objective function to control overfitting. XGBoost has several advantages over other algorithms. It can handle high-dimensional data and nonlinear relationships between features and targets. It is also less prone to overfitting compared to other ensemble learning algorithms. Additionally, it is computationally efficient and scalable, making it suitable for large datasets. XGBoost has been used in many applications, including recommendation systems, image classification, and financial analysis. It has won numerous machine learning competitions and is widely regarded as a state-of-the- art algorithm in the field of machine learning. In summary, XGBoost is an advanced version of the gradient boosting algorithm that uses several enhancements to improve model accuracy and prevent overfitting. It is a powerful algorithm that is widely used in many applications and has achieved impressive results in machine learning competitions.

CONCLUSION AND FUTURE WORK

Hence, this research work provides good results and can be used in predicting duplicate questions for study purposes. However, few complications like, extraction of many features and vectors, heavy use of memory by .csv file or any other file has to be taken care in future work .Due to memory issues it is difficult to load and save any changes every single time. Therefore, it is better to use "pickle" form of a file for efficient use of data. In order to reduce the risk of "Data Leakage" the data can be split and be used before training the models. To obtain the best parameter rather an implementing a random parameter for the models it is suggested to use "Grid search CV" or "Random search CV". Furthermore, "XG Boost" can be utilized to provide most accurate output, in real time problem solving.

This study uses Machine Learning and Natural Language Processing to classify whether question pairings are duplicates or not in Q&A forums. The use of minimal cost architecture and the selection of highly dominating elements from the questions make it an effective template for detecting duplicate inquiries and subsequently finding high-quality answers

REFERENCES

1. Broder, A. (1997) On the resemblance and containment of documents. Proceedings of the

Compression and Complexity of Sequences 1997, SEQUENCES’97, Washington, DC, USA. IEEE Computer Society.

2. Yoon Kim. (2014) Convolution neural networks for sentence classification. Proceedings of the 2015 Conference on Empirical Methods for Natural Language Processing, pages 1746-1751. Doha, Qatar.

3. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. (2013). Efficient estimation of word representations in vector space. Proceedings of International Conference on Learning Representations, ICLR 2013, Scottsdale, AZ, USA.

4. https://www.kaggle.com/c/quora-question-pairs

5. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

6. https://www.tensorflow.org/tutorials/word2ve

7. https://code.google.com/archive/p/word2vec/

8. http://machinelearningmastery.com/sequence-classification-lstmrecurrent-neural-networks-python-keras/

9. Hua He and Jimmy Lin. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In NAACL, 2016

10. P.A. Jadhav, P. N. Chatur and K. P. Wagh, "Integrating performance of web search engine with Machine Learning approach," 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio- Informatics (AEEICB), 2016, pp. 519-524.

11. P. P. Shelke and K. P. Wagh, "Review on Aspect based Sentiment Analysis on Social Data," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021, pp. 331-336.

12. Ms. Vishwaja M. Tambakhe, Dr. Kishor P.Wagh, "Review on Exploring Similarity between Two Questions Using Machine Learning", International Journal of Scientific Research in Computer Science, Engineering and Information Technology

13. http://www.erogol.com/duplicate-question-detection-deep-learning/

14. https://www.linkedin.com/pulse/duplicate-quora-question-abhishekth akur

15. Eneko Agirre et al. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity

16. Lei Yu et al. 2014. Deep Learning for Answer Sentence Selection

17. Mikhail Bilenko et al. 2003. Adaptive Duplicate Detection Using Learnable String Similarity Measures